

TUGAS AKHIR - KS 1501

**RANCANG BANGUN APLIKASI POINT OF SALE  
BERBASIS WEB RESPONSIF UNTUK PONSEL PINTAR**  
***DEVELOPING RESPONSIVE WEB BASED POINT OF  
SALE APPLICATION FOR SMARTPHONE***

BAGUS ARGA PUTRA  
NRP 05211440000005

Dosen Pembimbing  
Faizal Johan Atletiko S.Kom., M.T.

DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018

*Halaman ini sengaja dikosongkan*

**TUGAS AKHIR – KS 141501**

# **RANCANG BANGUN APLIKASI POINT OF SALE BERBASIS WEB RESPONSIF UNTUK PONSEL PINTAR**

**Bagus Arga Putra**  
**05211440000005**

**Dosen Pembimbing**  
**Faizal Johan Atletiko S.Kom., M.T.**

**DEPARTEMEN SISTEM INFORMASI**  
**Fakultas Teknologi Informasi dan Komunikasi**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2018**

*Halaman ini sengaja dikosongkan*



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**UNDERGRADUATE THESIS – KS 141501**

# ***DEVELOPING RESPONSIVE WEB BASED POINT OF SALE APPLICATION FOR SMARTPHONE***

**Bagus Arga Putra**

**05211440000005**

**Supervisor**

**Faizal Johan Atletiko S.Kom., M.T.**

**INFORMATION SYSTEMS DEPARTMENT**

**Information and Communications Technology Faculty**

**Sepuluh Nopember Institut of Technology**

**Surabaya 2018**

*Halaman ini sengaja dikosongkan*

## LEMBAR PENGESAHAN

### **RANCANG BANGUN APLIKASI *POINT OF SALE* BERBASIS WEB RESPONSIF UNTUK PONSEL PINTAR**

#### **TUGAS AKHIR**

**Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada**

**Departemen Sistem Informasi  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember**

Oleh:

**BAGUS ARG A PUTRA**  
**NRP. 0521144000005**

Surabaya, Juli 2018

**KEPALA  
DEPARTEMEN SISTEM INFORMASI**

**Dr. Ir. Aris Triahyanto, M.Kom**  
**NIP. 19650310 199102 1 001**







## LEMBAR PERSETUJUAN

### RANCANG BANGUN APLIKASI *POINT OF SALE* BERBASIS WEB RESPONSIF UNTUK PONSEL PINTAR

#### TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Departemen Sistem Informasi  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**BAGUS ARGA PUTRA**  
**NRP. 0521144000005**

Disetujui Tim Penguji : Tanggal Ujian : 10 Juli 2018  
Periode Wisuda : September 2018


**Faizal Johan Atletiko, S.Kom., M.Kom**

  
**(Pembimbing I)**

**Radityo Prasetyanto W., S.Kom., M.Kom**

  
**(Penguji I)**

**Renny Pradina K., S.T, M.T., SCJP**

  
**(Penguji II)**

*Halaman ini sengaja dikosongkan*



# **RANCANG BANGUN APLIKASI *POINT OF SALE* BERBASIS *WEB RESPONSIF* UNTUK PONSEL PINTAR**

Nama Mahasiswa : Bagus Arga Putra  
NRP : 05211440000005  
Departemen : Sistem Informasi FTIK  
Pembimbing I : Faizal Johan Atletiko S.Kom., M.T.

## **ABSTRAK**

*Point of sale merupakan sistem yang menangani transaksi jual beli dan pembayaran, sistem POS juga membantu dalam memberikan laporan penjualan dan stok barang. Sistem POS yang dapat berjalan tidak hanya pada satu jenis perangkat memudahkan penggunaan POS untuk digunakan di perangkat mana saja, kapan saja, dan diman saja. UD Intan Jaya merupakan sebuah toko di Tulungagung yang memiliki toko cabang dan sering membuka stand di pameran dan bazar, beberapa masalah yang dihadapi adalah seperti mobilitas dalam menggunakan mesin kasir di tempat-tempat berbeda dan pelaporan penjualan dari toko cabang dan stand-stand untuk toko utama yang tidak terintegrasi.*

*Dengan perancangan sistem point of sale berbasis web yang responsif dan dapat diakses melalui perangkat mobile, diharapkan dapat membantu UD Intan Jaya dalam menjalankan transaksi jual beli mereka di toko cabang mereka dan stand yang dibuka di pameran atau bazar untuk memberikan kemudahan dalam melakukan akses POS untuk melakukan transaksi jual beli mereka dan memudahkan dalam memberikan laporan penjualan dan stok barang pada masing-masing toko kepada toko utama.*

***Kata Kunci: point of sale, mobilitas, transaksi, pelaporan***

*Halaman ini sengaja dikosongkan*

# **DEVELOPING RESPONSIVE WEB BASED POINT OF SALE APPLICATION FOR SMARTPHONE**

Student Name : Bagus Arga Putra  
NRP : 05211440000005  
Department : Information Systems FTIK  
Supervisor I : Faizal Johan Atletiko S.Kom., M.T.

## **ABSTRACT**

*Point of sale is a system that handles buy and sell transactions, POS system also helps in providing reports on sales and stock of goods. A POS system that can run not just on one device type makes it easy to use POS for use on any device, anytime, anywhere. UD Intan Jaya is a shop in Tulungagung which has a branch shop and often open booths at fairs and bazaars, some of the problems faced are such as mobility in using cash register machines in different places and sales reporting from branch shops and booths for main shops which is not integrated.*

*With the design of a web-based point of sale system that is responsive and accessible via mobile devices, it is expected to assist UD Intan Jaya in running their buying and selling transactions at their branches and booths opened in exhibitions or bazaars to provide easy access to POS to do buy and sell transactions and facilitate in providing reports on the sale and stock of goods in each store to the main store.*

***Keywords: point of sale, mobility, transaction, reporting***

*Halaman ini sengaja dikosongkan*



## KATA PENGANTAR

Puji syukur penulis ucapkan kepada Allah SWT atas rahmat, taufik dan hidayah-Nya sehingga penulis dapat menyelesaikan buku Tugas Akhir yang berjudul “Rancang Bangun Aplikasi *Point of Sale* Berbasis Web Responsif untuk Ponsel Pintar” dengan tepat waktu

Pada kesempatan ini penulis juga ingin mengucapkan terima kasih kepada beberapa pihak yang telah memberikan banyak hal, baik berupa bantuan, dukungan, bimbingan, arahan, kasih sayang, semangat kepada penulis, sehingga tugas akhir ini bisa selesai tepat pada waktunya. Beberapa pihak tersebut yaitu:

1. Kedua Orang Tua penulis, yang selalu berdoa dan bekerja keras tanpa henti demi kelancaran dan kesuksesan penulis hingga sampai saat ini.
2. Bapak Faizal Johan Atletiko S.Kom., M.T. selaku dosen pembimbing penulis yang membantu memberikan ide, dan senantiasa memberikan bimbingan, saran, kritik, dan arahan dalam menyusun buku tugas akhir ini.
3. Bapak Achmad Holil Noor Ali, Ir., M.Kom. selaku dosen wali dari penulis yang membantu dalam masa masa perkuliahan terutama saat FRS.
4. Seluruh dosen Departemen Sistem Informasi ITS yang telah memberikan ilmu pengetahuan dan pengalaman yang sangat berharga dan bermanfaat bagi penulis.
5. Saudara dekat penulis, yang telah membantu penulis ketika masih berusaha sebelum kuliah hingga penulis kuliah yang telah banyak direpotkan oleh penulis.
6. Teman-teman grup S/k Nogo yang merupakan teman terdekat penulis saat masa perkuliahan yang selalu menemani dan membantu dalam masa-masa perkuliahan
7. Teman-teman Kaporits maupun kontrakan ungu yang selalu menemani penulis ketika melakukan pembuatan buku tugas akhir ini selama berada di kontrakan
8. Teman-teman seperjuangan Lab ADDI dan OSIRIS, yang menemani masa perkuliahan, dan telah menjadi keluarga penulis selama berada di Surabaya.

9. Pada seluruh pihak yang telah membantu menyelesaikan tugas akhir ini baik secara langsung maupun tak langsung yang tidak bisa penulis cantumkan satu per satu.

Kesempurnaan adalah hanya milik Allah SWT semata, sehingga penulis menyadari bahwa memiliki kekurangan yang ada pada buku tugas akhir ini, Sehingga penulis memohon maaf dan akan menerima segala bentuk kritik dan saran yang diberikan demi memperbaiki buku tugas akhir ini. Akhir kata semoga buku tugas akhir ini bisa memberikan manfaat kepada seluruh pembaca dan memberikan kontribusi kepada ilmu pengetahuan terutama penelitian terkait.

## DAFTAR ISI

ABSTRAK .....	i
ABSTRACT .....	iii
KATA PENGANTAR .....	v
DAFTAR ISI .....	vii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xvii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	3
1.5.1 Bagi Perusahaan .....	3
1.5.2 Bagi Akademik .....	3
1.6 Relevansi .....	3
BAB II TINJAUAN PUSTAKA .....	5
2.1 Produk Sejenis .....	5
2.1.1 Olsera POS .....	5
2.1.2 Kounta Mobile POS System .....	5
2.1.3 MOKA Point of Sale .....	5
2.2 Penelitian Sebelumnya .....	6
2.2.1 Pembuatan Aplikasi Point of sale Toko Cabang Perusahaan Torani Menggunakan Framework Codeigniter	6
2.2.2 Pembuatan Aplikasi Point of sales untuk Rumah Makan Dapur Rinjani .....	6
2.2.3 Penerapan Teknologi Responsive Web Design menggunakan Library Bootstrap Untuk Pembuatan Aplikasi E-Journal pada Yayasan Bina Darma Salatiga	7
2.3 Dasar Teori .....	7
2.3.1 Point of sales .....	7
2.3.2 Web Responsif .....	8
2.3.3 Bootstrap .....	9

2.3.4	Android Web Server .....	11
BAB III METODOLOGI PENELITIAN .....		13
3.1	Tahapan Tugas Akhir .....	13
3.1.1	Studi Literatur.....	14
3.1.2	Analisa Studi Kasus.....	14
3.1.3	Analisa Kebutuhan .....	14
3.1.4	Implementasi .....	15
3.1.5	Penyusunan Laporan.....	16
3.2	Rancangan Arsitektur Sistem .....	16
3.3	Jadwal Kegiatan .....	17
BAB IV PERANCANGAN .....		19
4.1	<i>Requirement Review</i> .....	19
4.1.1	Analisa Kebutuhan .....	19
4.1.2	Domain Model.....	20
4.1.3	Use Case.....	21
4.2	<i>Preliminary Design Review</i> .....	33
4.2.1	Robustness Analysis .....	33
4.2.2	Updated Domain Model .....	44
4.3	<i>Detailed Design Review</i> .....	44
4.3.1	Sequence Diagram.....	45
4.3.2	Class Diagram .....	58
4.4	Desain Basis Data.....	60
4.5	Rancangan Desain Antarmuka.....	61
BAB V IMPLEMENTASI .....		69
5.1.	Lingkungan Implementasi .....	69
5.2.	Konfigurasi Server.....	70
5.3.	Pembuatan Aplikasi Point of Sale.....	70
5.3.1.	Fungsi Login.....	71
5.3.2.	Fungsi Registrasi .....	73
5.3.3.	Fungsi Input Produk Baru.....	76
5.3.4.	Fungsi Edit Informasi Produk.....	78
5.3.5.	Fungsi Menghapus Produk .....	81
5.3.6.	Fungsi Input Pegawai Baru.....	82
5.3.7.	Fungsi Edit Informasi Pegawai .....	83
5.3.8.	Fungsi Menghapus Pegawai .....	86

5.3.9. Point of Sale: Fungsi Menambahkan Barang ke Keranjang Belanja.....	87
5.3.10. Point of Sale: Menghapus Barang dari Keranjang Belanja.....	92
5.3.11. Point of Sale: Fungsi Menambahkan Diskon.....	94
5.3.12. Point of Sale: Fungsi Pembayaran .....	99
5.3.13. Point of Sale: Pengaturan Gambar .....	106
5.3.14. Point of Sale: Pencarian Produk .....	108
5.3.15. Point of Sale: Filter Kategori Produk.....	110
5.3.16. Fungsi Menampilkan Data Penjualan .....	111
5.3.17. Fungsi Sinkronisasi Data Penjualan.....	112
5.4. Pengujian Aplikasi .....	116
5.4.1. Pengujian Fungsional .....	116
5.4.2. Pengujian Kompatibilitas .....	116
<b>BAB VI HASIL DAN PEMBAHASAN .....</b>	<b>117</b>
6.1. Hasil .....	117
6.1.1. Pengujian Fungsional Aplikasi .....	117
6.1.2. Pengujian Kompatibilitas Layar .....	118
6.1.3. Kesimpulan Uji Coba .....	125
<b>BAB VII KESIMPULAN DAN SARAN.....</b>	<b>127</b>
7.1. Kesimpulan .....	127
7.2. Saran.....	127
<b>DAFTAR PUSTAKA .....</b>	<b>129</b>
<b>LAMPIRAN.....</b>	<b>131</b>
A. Tabel Pengujian Fungsional .....	131

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

<i>Gambar 3.1 Alur Pengerjaan Tugas Akhir .....</i>	<i>13</i>
<i>Gambar 3.2 Rancangan Arsitektur Sistem POS.....</i>	<i>16</i>
<i>Gambar 4.1 Domain model awal berdasarkan kebutuhan fungsional .....</i>	<i>21</i>
<i>Gambar 4.2 Diagram use case dari sistem .....</i>	<i>23</i>
<i>Gambar 4.3 Robustness diagram fungsi Login .....</i>	<i>33</i>
<i>Gambar 4.4 Robustness diagram fungsi Registrasi .....</i>	<i>34</i>
<i>Gambar 4.5 Robustness diagram fungsi Input Produk Baru.....</i>	<i>35</i>
<i>Gambar 4.6 Robustness diagram fungsi Edit Informasi Produk.</i>	<i>36</i>
<i>Gambar 4.7 Robustness diagram fungsi Menghapus Produk .....</i>	<i>37</i>
<i>Gambar 4.8 Robustness diagram fungsi Input Pegawai Baru ....</i>	<i>37</i>
<i>Gambar 4.9 Robustness diagram fungsi Edit Informasi Pegawai .....</i>	<i>38</i>
<i>Gambar 4.10 Robustness diagram fungsi Menghapus Pegawai .</i>	<i>39</i>
<i>Gambar 4.11 Robustness diagram fungsi Menambahkan Barang ke Keranjang Belanja.....</i>	<i>40</i>
<i>Gambar 4.12 Robustness diagram fungsi Menghapus Barang dari Keranjang Belanja.....</i>	<i>40</i>
<i>Gambar 4.13 Robustness diagram fungsi tambah diskon .....</i>	<i>41</i>
<i>Gambar 4.14 Robustness diagram fungsi Menampilkan Data Penjualan.....</i>	<i>42</i>
<i>Gambar 4.15 Robustness diagram fungsi Pembayaran .....</i>	<i>42</i>
<i>Gambar 4.16 Gambar 4.17 Robustness diagram fungsi Menampilkan Data Penjualan .....</i>	<i>43</i>
<i>Gambar 4.18 Robustness diagram fungsi Sinkronisasi Basis Data .....</i>	<i>43</i>
<i>Gambar 4.19 Domain model dengan atribut .....</i>	<i>44</i>
<i>Gambar 4.20 Sequence Diagram fungsi Login .....</i>	<i>45</i>
<i>Gambar 4.21 Sequence Diagram fungsi Registrasi .....</i>	<i>46</i>
<i>Gambar 4.22 Sequence Diagram fungsi Input Produk Baru.....</i>	<i>47</i>
<i>Gambar 4.23 Sequence Diagram fungsi Edit Informasi Produk.</i>	<i>48</i>
<i>Gambar 4.24 Sequence Diagram fungsi Menghapus Produk .....</i>	<i>49</i>
<i>Gambar 4.25 Input Pegawai Baru .....</i>	<i>50</i>

<i>Gambar 4.26 Sequence Diagram fungsi Edit Informasi Pegawai</i>	51
<i>Gambar 4.27 Sequence Diagram fungsi Menghapus Pegawai</i>	52
<i>Gambar 4.28 Sequence Diagram fungsi Memasukkan Barang ke Keranjang Belanja</i>	53
<i>Gambar 4.29 Sequence Diagram fungsi Menghapus Barang dari Keranjang Belanja</i>	54
<i>Gambar 4.30 Sequence Diagram fungsi Tambah Diskon</i>	55
<i>Gambar 4.31 Sequence Diagram fungsi Pembayaran</i>	56
<i>Gambar 4.32 Sequence Diagram fungsi menampilkan data penjualan</i>	57
<i>Gambar 4.33 Sequence Diagram fungsi Sinkronisasi Data</i>	58
<i>Gambar 4.34 Class diagram dari sistem</i>	59
<i>Gambar 4.35 Skema basis data aplikasi point of sale</i>	60
<i>Gambar 4.36 Rancangan desain antarmuka halaman Login</i>	62
<i>Gambar 4.37 Rancangan desain antarmuka halaman Registrasi</i>	62
<i>Gambar 4.38 Rancangan desain antarmuka halaman Point of Sale</i>	63
<i>Gambar 4.39 Rancangan desain antarmuka modal pop-up memasukkan jumlah barang</i>	63
<i>Gambar 4.40 Rancangan desain antarmuka modal pop-up konfirmasi hapus barang</i>	64
<i>Gambar 4.41 Rancangan desain antarmuka modal pop-up memasukkan jumlah pembayaran</i>	64
<i>Gambar 4.42 Rancangan desain antarmuka halaman daftar produk</i>	65
<i>Gambar 4.43 Rancangan desain antarmuka halaman Input Produk Baru</i>	66
<i>Gambar 4.44 Rancangan desain antarmuka halaman Edit Informasi Produk</i>	66
<i>Gambar 4.45 Rancangan desain antarmuka halaman Daftar Pegawai</i>	67
<i>Gambar 4.46 Rancangan desain antarmuka halaman Edit Informasi Pegawai</i>	68
<i>Gambar 4.47 Rancangan desain antarmuka halaman Input Pegawai Baru</i>	68



<i>Gambar 5.1 Tampilan halaman Login.....</i>	<i>71</i>
<i>Gambar 5.2 Kode pemrograman fungsi Login.....</i>	<i>72</i>
<i>Gambar 5.3 Kode pemrograman jika pengguna sudah melakukan login.....</i>	<i>73</i>
<i>Gambar 5.4 Tampilan halaman Registrasi .....</i>	<i>74</i>
<i>Gambar 5.5 Kode query select Username dan Email pengguna. 74</i>	
<i>Gambar 5.6 Kode verifikasi dua password .....</i>	<i>75</i>
<i>Gambar 5.7 Kode jika username atau email sudah ada.....</i>	<i>75</i>
<i>Gambar 5.8 Query memasukkan data pengguna untuk didaftarkan .....</i>	<i>76</i>
<i>Gambar 5.9 Tampilan halaman input produk baru .....</i>	<i>77</i>
<i>Gambar 5.10 Kode untuk mendapatkan input file gambar .....</i>	<i>77</i>
<i>Gambar 5.11 Kode query memasukkan data produk ke basis data .....</i>	<i>78</i>
<i>Gambar 5.12 Lokasi tombol Edit pada daftar produk .....</i>	<i>78</i>
<i>Gambar 5.13 Tampilan halaman Edit Produk.....</i>	<i>79</i>
<i>Gambar 5.14 Kode untuk memperbarui data yang diedit jika gambar diinputkan.....</i>	<i>80</i>
<i>Gambar 5.15 Kode memperbarui data yang diedit tanpa input gambar.....</i>	<i>80</i>
<i>Gambar 5.16 Pop-up konfirmasi hapus .....</i>	<i>81</i>
<i>Gambar 5.17 Kode query hapus produk yang dipilih .....</i>	<i>81</i>
<i>Gambar 5.18 Tampilan halaman Tambah Pegawai .....</i>	<i>82</i>
<i>Gambar 5.19 Kode untuk mendapatkan input file gambar .....</i>	<i>83</i>
<i>Gambar 5.20 Kode query memasukkan data produk ke basis data .....</i>	<i>83</i>
<i>Gambar 5.21 Tampilan halaman Edit Pegawai.....</i>	<i>84</i>
<i>Gambar 5.22 Lokasi tombol Edit pada daftar produk .....</i>	<i>84</i>
<i>Gambar 5.23 Kode untuk memperbarui data yang diedit jika gambar diinputkan.....</i>	<i>85</i>
<i>Gambar 5.24 Kode untuk memperbarui data yang edit tanpa input gambar.....</i>	<i>85</i>
<i>Gambar 5.25 Pop-up konfirmasi hapus .....</i>	<i>86</i>
<i>Gambar 5.26 Kode query hapus pegawai yang dipilih .....</i>	<i>86</i>
<i>Gambar 5.27 Tampilan halaman Point of Sale.....</i>	<i>87</i>
<i>Gambar 5.28 Pop-up input jumlah barang .....</i>	<i>88</i>

<i>Gambar 5.29 Kode fungsi spinner.....</i>	<i>88</i>
<i>Gambar 5.30 Kode fungsi tombol input jumlah.....</i>	<i>89</i>
<i>Gambar 5.31 Kode definisi variabel yang digunakan .....</i>	<i>89</i>
<i>Gambar 5.32 Kode query untuk mencari barang yang dimasukkan .....</i>	<i>90</i>
<i>Gambar 5.33 Kode query memperbarui data yang sudah ada ....</i>	<i>90</i>
<i>Gambar 5.34 Kode query memasukkan data barang.....</i>	<i>90</i>
<i>Gambar 5.35 Kode setelah dilakukan post.....</i>	<i>91</i>
<i>Gambar 5.36 Kode fungsi reload .....</i>	<i>91</i>
<i>Gambar 5.37 Tampilan keranjang belanja setelah dimasukkan produk.....</i>	<i>92</i>
<i>Gambar 5.38 Pop-up konfirmasi hapus barang .....</i>	<i>93</i>
<i>Gambar 5.39 Kode tombol Iya (hapus) .....</i>	<i>93</i>
<i>Gambar 5.40 Kode query hapus barang .....</i>	<i>94</i>
<i>Gambar 5.41 Kode setelah sukses dihapus .....</i>	<i>94</i>
<i>Gambar 5.42 Pop-up input diskon .....</i>	<i>95</i>
<i>Gambar 5.43 Pilihan jenis / satuan diskon.....</i>	<i>95</i>
<i>Gambar 5.44 Input diskon dengn satuan persen .....</i>	<i>96</i>
<i>Gambar 5.45 Kode untuk tombol angka pada satuan rupiah .....</i>	<i>96</i>
<i>Gambar 5.46 Kode untuk tombol angka pada satuan persen .....</i>	<i>97</i>
<i>Gambar 5.47 Kode untuk post informasi diskon .....</i>	<i>97</i>
<i>Gambar 5.48 Kode query untuk mendapatkan data transaksi.....</i>	<i>98</i>
<i>Gambar 5.49 Kode query untuk memperbarui data .....</i>	<i>98</i>
<i>Gambar 5.50 Kode query untuk memasukkan data diskon.....</i>	<i>98</i>
<i>Gambar 5.51 Tampilan informasi diskon pada keranjang belanja .....</i>	<i>99</i>
<i>Gambar 5.52 Kode setelah memasukkan informasi diskon .....</i>	<i>99</i>
<i>Gambar 5.53 Pop-up input pembayaran.....</i>	<i>100</i>
<i>Gambar 5.54 Kode untuk tombol bayar .....</i>	<i>100</i>
<i>Gambar 5.55 Kode query untuk mendapatkan data transaksi...101</i>	<i>101</i>
<i>Gambar 5.56 Kode query memperbarui data transaksi.....101</i>	<i>101</i>
<i>Gambar 5.57 Kode query memasukkan data pembayaran .....</i>	<i>101</i>
<i>Gambar 5.58 Kode setelah memasukkan data pembayaran .....</i>	<i>102</i>
<i>Gambar 5.59 Tampilan informasi pembayaran pada keranjang belanja .....</i>	<i>102</i>
<i>Gambar 5.60 Kode untuk tombol selesai.....103</i>	<i>103</i>

<i>Gambar 5.61 Kode definisi variabel yang digunakan.....</i>	<i>103</i>
<i>Gambar 5.62 Kode menentukan nilai berdasar satuan.....</i>	<i>103</i>
<i>Gambar 5.63 Kode mendefinisikan kode penjualan.....</i>	<i>104</i>
<i>Gambar 5.64 Kode query memasukkan data transaksi .....</i>	<i>104</i>
<i>Gambar 5.65 Kode query mendapatkan data di keranjang belanja .....</i>	<i>104</i>
<i>Gambar 5.66 Kode query memasukkan informasi detail penjualan .....</i>	<i>105</i>
<i>Gambar 5.67 Kode query menghapus data sementara .....</i>	<i>105</i>
<i>Gambar 5.68 Kode setelah proses selesai dilakukan .....</i>	<i>105</i>
<i>Gambar 5.69 Lokasi tombol toggle.....</i>	<i>106</i>
<i>Gambar 5.70 Kode fungsi tombol toggle .....</i>	<i>106</i>
<i>Gambar 5.71 Kode mengeset setting.....</i>	<i>107</i>
<i>Gambar 5.72 Tampilan dengan tanpa gambar .....</i>	<i>107</i>
<i>Gambar 5.73 Tampilan dengan gambar.....</i>	<i>108</i>
<i>Gambar 5.74 Kode fungsi pencarian produk.....</i>	<i>108</i>
<i>Gambar 5.75 Tampilan filter pencarian sesuai input .....</i>	<i>109</i>
<i>Gambar 5.76 Kode fungsi tombol kategori.....</i>	<i>110</i>
<i>Gambar 5.77 Tampilan setelah tombol kategori diklik .....</i>	<i>111</i>
<i>Gambar 5.78 Tampilan halaman daftar penjualan.....</i>	<i>111</i>
<i>Gambar 5.79 Tampilan halaman detail penjualan .....</i>	<i>112</i>
<i>Gambar 5.80 Tampilan halaman sinkronisasi .....</i>	<i>112</i>
<i>Gambar 5.81 Kode jika memilih hari ini .....</i>	<i>113</i>
<i>Gambar 5.82 Kode jika memilih minggu ini .....</i>	<i>113</i>
<i>Gambar 5.83 Kode jika memilih tanggal .....</i>	<i>114</i>
<i>Gambar 5.84 Kode query mendapatkan data sesuai klausa.....</i>	<i>114</i>
<i>Gambar 5.85 Kode memasukkan data ke array .....</i>	<i>115</i>
<i>Gambar 5.86 Kode menutup dan membuka koneksi basis data .....</i>	<i>115</i>
<i>Gambar 5.87 Kode memasukkan array ke basis data .....</i>	<i>115</i>
<i>Gambar 6.1 Tampilan daftar barang pada Android versi 5.0... ..</i>	<i>119</i>
<i>Gambar 6.2 Tampilan daftar barang pada Android versi 6.0... ..</i>	<i>120</i>
<i>Gambar 6.3 Tampilan daftar barang pada Android versi 7.0... ..</i>	<i>120</i>
<i>Gambar 6.4 Tampilan keranjang belanja pada Android versi 5.0 .....</i>	<i>122</i>
<i>Gambar 6.5 Tampilan keranjang belanja pada Android versi 7.0 .....</i>	<i>123</i>

*Gambar 6.6 Tampilan keranjang belanja pada Android versi 6.0*  
.....123

*Gambar 6.7 Tampilan modal* .....125

## DAFTAR TABEL

<i>Tabel 3.1 Jadwal Pengerjaan Tugas Akhir .....</i>	<i>17</i>
<i>Tabel 4.1 Daftar Use Case dalam sistem .....</i>	<i>22</i>
<i>Tabel 4.2 Use case description fungsi login.....</i>	<i>23</i>
<i>Tabel 4.3 Use case description fungsi Registrasi.....</i>	<i>24</i>
<i>Tabel 4.4 Use case description fungsi Logout.....</i>	<i>25</i>
<i>Tabel 4.5 Use case description fungsi Input Produk Baru .....</i>	<i>25</i>
<i>Tabel 4.6 Use case description fungsi Edit Produk.....</i>	<i>26</i>
<i>Tabel 4.7 Use case description fungsi Hapus Produk.....</i>	<i>26</i>
<i>Tabel 4.8 Use case description fungsi Menambah Pegawai Baru .....</i>	<i>27</i>
<i>Tabel 4.9 Use case description fungsi Edit Pegawai .....</i>	<i>27</i>
<i>Tabel 4.10 Use case description fungsi Hapus Pegawai.....</i>	<i>28</i>
<i>Tabel 4.11 Use case description fungsi Memasukkan Barang ke Keranjang Belanja.....</i>	<i>28</i>
<i>Tabel 4.12 Use case description fungsi Hapus Barang dari Keranjang Belanja.....</i>	<i>29</i>
<i>Tabel 4.13 Use case description fungsi Menambah Diskon .....</i>	<i>29</i>
<i>Tabel 4.14 Use case description fungsi Pembayaran.....</i>	<i>30</i>
<i>Tabel 4.15 Use case description fungsi Menampilkan Data Penjualan.....</i>	<i>31</i>
<i>Tabel 4.16 Use case description fungsi Sinkronisasi Data .....</i>	<i>31</i>
<i>Tabel 5.1 Spesifikasi komputer yang digunakan .....</i>	<i>69</i>
<i>Tabel 5.2 Spesifikasi perangkat Android yang digunakan .....</i>	<i>69</i>
<i>Tabel 5.3 Teknologi yang digunakan .....</i>	<i>70</i>
<i>Tabel 5.4 Format test case.....</i>	<i>116</i>
<i>Tabel 6.1 Hasil pengujian fungsional .....</i>	<i>117</i>
<i>Tabel 9.1 Test Case Login .....</i>	<i>131</i>
<i>Tabel 9.2 Test Case Registrasi.....</i>	<i>131</i>
<i>Tabel 9.3 Test Case Input Produk Baru.....</i>	<i>132</i>
<i>Tabel 9.4 Test Case Edit Informasi Produk .....</i>	<i>133</i>
<i>Tabel 9.5 Test Case Menghapus Produk.....</i>	<i>133</i>
<i>Tabel 9.6 Test Case Input Pegawai Produk .....</i>	<i>134</i>
<i>Tabel 9.7 Test Case Edit Informasi Pegawai .....</i>	<i>134</i>

<i>Tabel 9.8 Test Case Menghapus Pegawai .....</i>	<i>135</i>
<i>Tabel 9.9 Test Case Menambah Barang ke Keranjang Belanja .....</i>	<i>135</i>
<i>Tabel 9.10 Test Case Menghapus Barang dari Keranjang Belanja .....</i>	<i>136</i>
<i>Tabel 9.11 Test Case Menambah Diskon .....</i>	<i>136</i>
<i>Tabel 9.12 Test Case Pembayaran .....</i>	<i>137</i>
<i>Tabel 9.13 Test Case Pengaturan Gambar .....</i>	<i>138</i>
<i>Tabel 9.14 Test Case Pencarian Produk .....</i>	<i>138</i>
<i>Tabel 9.15 Test Case Filter Kategori Produk .....</i>	<i>139</i>
<i>Tabel 9.16 Test Case Menampilkan Data Penjualan .....</i>	<i>140</i>
<i>Tabel 9.17 Test Case Sinkronisasi Data Penjualan .....</i>	<i>140</i>

# **BAB I**

## **PENDAHULUAN**

Pada bab ini akan dijelaskan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat serta relevansi tugas akhir.

### **1.1 Latar Belakang**

*Point of sale* atau POS merupakan sistem yang menangani proses transaksi jual beli dan pelanggan melakukan pembayaran atas barang atau jasa kepada penjual, atau dengan kata lain *point of sale* adalah sistem kasir. Sistem POS lebih dari sekedar menangani transaksi pada umumnya, seperti penggunaan kartu kredit, menangani sistem persediaan, dan pelaporan penjualan. Sistem POS biasanya diintegrasikan dengan layanan lainnya untuk mempermudah proses manajemen bisnis. Dengan menggunakan sistem POS, toko atau perusahaan akan mendapatkan banyak keuntungan selain hanya mempermudah proses transaksi, seperti dapat memberikan pelaporan terperinci untuk setiap penjualan yang dilakukan, perhitungan persediaan barang yang dapat dengan otomatis diperbarui pada tiap penjualan yang dilakukan, tersedianya data yang akurat dan mudah diakses [1].

UD Intan Jaya merupakan sebuah UMKM di Tulungagung yang menjual jajanan khas Tulungagung sebagai oleh-oleh. UD Intan Jaya memiliki beberapa cabang yang juga terletak di daerah Tulungagung, tidak hanya cabang, UD Intan Jaya juga kerap membuka stand di acara-acara seperti pameran dan bazaar yang biasa diadakan di Tulungagung. Dengan kerapnya mereka membuka stand di beberapa acara, mereka pasti membutuhkan sistem kasir yang mudah digunakan dan dipindahkan ke satu tempat ke tempat lain, apalagi jika tempat tersebut tidak cukup luas dan hanya menyediakan sumber listrik yang tidak besar.

Dengan masalah tersebut, pada tugas akhir ini ditawarkan solusi berupa *point of sales* berbasis *web* yang responsif sehingga dapat diakses dengan menggunakan perangkat *mobile smartphone* dan tablet, dengan perkiraan ukuran minimum layar ponsel adalah

5.5 inch. Tujuan dari pembuatan aplikasi *point of sale* ini adalah untuk membantu UD Intan Jaya dalam melakukan aktivitas jual beli mereka yang membutuhkan mobilitas dalam membuka beberapa stand, dimana mereka akan dapat dengan mudah menempatkan sistem POS di stand pameran dan bazaar dan memungkinkan untuk ditempatkan pada tempat yang tidak cukup luas dan tidak ada sumber listrik yang besar. Selain itu ini juga memudahkan pihak UMKM jika hendak membuka cabang baru yang mana tidak perlu membeli mesin kasir baru atau menggunakan sistem kasir manual. Karena POS ini berbasis *web*, maka pihak UMKM tidak perlu melakukan download dan cukup membuka aplikasi POS secara online melalui *web browser*, yang mana juga memudahkan integrasi dengan cabang lain dan stand yang dibuka dengan pusat, sehingga data *inventory* bisa selalu diupdate setiap terjadi transaksi melalui POS, serta dapat memberikan pelaporan penjualan dengan mendetail sesuai transaksi yang ada.

## **1.2 Rumusan Masalah**

1. Bagaimana pembuatan aplikasi *point of sale* yang responsif pada UD Intan Jaya?
2. Bagaimana pembuatan aplikasi *point of sale* yang fleksibel untuk UD Intan Jaya?

## **1.3 Batasan Masalah**

Dari permasalahan yang disebutkan di atas, batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Penelitian dilakukan untuk penerapan aplikasi pada UD Intan Jaya.
2. Hasil akhir dari penelitian ini adalah pengembangan aplikasi berbasis *web* responsif yang sesuai digunakan pada perangkat *mobile*.
3. Target dari aplikasi web POS ini adalah dapat digunakan untuk sekitar 5 perangkat dalam satu jaringan lokal dan dihostingkan dengan penyedia hosting gratis.



#### **1.4 Tujuan Penelitian**

Tujuan dari penelitian ini adalah mengembangkan aplikasi *point of sales* berbasis *web* yang sesuai dengan perangkat *mobile* yang diterapkan pada UD Intan Jaya dalam membantu transaksi jual penjualan di kios-kios cabang dan stand yang dibuka di acara seperti bazar yang membutuhkan mobilitas dalam membawa dan menempatkan mesin kasir.

#### **1.5 Manfaat Penelitian**

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut.

##### **1.5.1 Bagi Perusahaan**

Dapat membantu UD Intan Jaya dalam melakukan transaksi jual beli yang berpindah-pindah tempat dengan mudah dengan cukup menggunakan perangkat *mobile* seperti *smartphone* atau tablet.

##### **1.5.2 Bagi Akademik**

Penelitian ini dapat dikembangkan lebih lanjut mengenai beberapa aspek yang belum ada pada penelitian ini untuk menjadi lebih baik.

#### **1.6 Relevansi**

Topik dari tugas akhir ini adalah rancang bangun aplikasi *point of sales* berbasis *web* untuk *mobile* dengan menggunakan *framework Bootstrap* untuk membuat *web* responsif. Tugas akhir ini memecahkan masalah dalam penempatan mesin kasir bagi perusahaan yang membutuhkan mobilitas dalam transaksi jual belinya, sehingga topik ini layak dijadikan sebagai tugas akhir untuk tingkat S1.

Tugas akhir ini berkaitan dengan mata kuliah Konstruksi dan Pengembangan Perangkat Lunak, Pemrograman Berbasis *Web*, dan Pengembangan Sumber Daya Perusahaan, sehingga layak sebagai tugas akhir dari Departemen Sistem Informasi.

*Halaman ini sengaja dikosongkan*

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini akan dijelaskan mengenai produk sejenis, penelitian sebelumnya, dan dasar teori yang akan dijadikan acuan atau landasan dalam pengerjaan tugas akhir ini

#### **2.1 Produk Sejenis**

Pada bagian ini akan dijelaskan mengenai produk sejenis yang sudah ada sebelumnya.

##### **2.1.1 Olsera POS**

Merupakan aplikasi *point of sale* dari Olsera, yang mendukung platform Windows, iOS, dan Android. Aplikasi POS dari Olsera ini cocok untuk toko ritel, mini market, dan restoran. Olsera POS mendukung integrasi aplikasi dengan perangkat *printer* struk dan juga Olsera menyediakan beberapa perangkat kasir dan *printer* yang mendukung aplikasi POS mereka. Beberapa fitur dari Olsera POS adalah sistem *cloud*, keamanan, laporan *real time*, *multi-platform*, manajemen persediaan dan hubungan pelanggan [2].

##### **2.1.2 Kounta Mobile POS System**

Aplikasi *point of sale* ini dikembangkan oleh perusahaan perangkat lunak Kounta, yang dapat berjalan di beberapa *platform*, iOS, Android, Mac, dan Windows. Aplikasi Kounta POS berjalan pada web browser sehingga tidak memerlukan instalasi aplikasi dan cocok untuk dibuka melalui ponsel iOS atau Android. Kounta POS juga dapat diintegrasikan dengan beberapa jenis perangkat seperti printer nota dan mesin kasir. Aplikasi ini juga tersedia fitur manajemen persediaan, manajemen hubungan pelanggan, pengawasan pegawai, pembuatan nota, dan pelaporan [3].

##### **2.1.3 MOKA Point of Sale**

Merupakan aplikasi *point of sale* yang dikembangkan oleh PT Moka Teknologi Indonesia. Berbeda dengan aplikasi POS sebelumnya, MOKA POS ini benar-benar aplikasi POS *mobile*, yang hanya tersedia untuk perangkat Android dan iOS. Fitur dari

MOKA POS hampir sama dengan dua aplikasi POS sebelumnya, seperti manajemen persediaan, manajemen pelanggan, pelaporan, manajemen pegawai, dan mendukung integrasi dengan mesin kasir. MOKA POS cocok untuk jenis usaha restoran / café, salon, dan ritel [4].

## **2.2 Penelitian Sebelumnya**

### **2.2.1 Pembuatan Aplikasi Point of sale Toko Cabang Perusahaan Torani Menggunakan Framework Codeigniter**

Penelitian ini berjudul “Pembuatan Aplikasi *Point of sale* Toko Cabang Perusahaan Torani Menggunakan *Framework Codeigniter*” [5] oleh Jodhi Sugihartono, Kodrat Iman Satoto, dan Eko Didik Widiyanto. Dalam penelitian ini dilakukan pembuatan aplikasi *point of sale* berbasis *web* untuk toko cabang perusahaan pemasok bahan olahan ikan mentah Torani Cirebon menggunakan *framework* CodeIgniter dan database MySQL . Aplikasi dirancang untuk membuat sistem pendataan barang yang dapat memudahkan pengecekan stok barang, sistem pemesanan barang di toko cabang untuk pemesanan barang yang habis di toko, dan sistem kasir untuk memudahkan proses transaksi jual beli di toko.

### **2.2.2 Pembuatan Aplikasi Point of sales untuk Rumah Makan Dapur Rinjani**

Penelitian ini berjudul “Pembuatan Aplikasi *Point of sales* untuk Rumah Makan Dapur Rinjani” [6] oleh Hendy Thomas Herman, Silvia Rostianingsih, dan Alexander Setiawan. Pada penelitian ini dibuat aplikasi pemesanan berbasis *web* yang berada pada jaringan lokal untuk Rumah Makan Dapur Rinjani Malang. Aplikasi *point of sale* ini dibuat menggunakan *framework* PHP Laravel, Javascript, HTML sebagai bahasa pemrograman pada *web* restoran, dan MySQL sebagai database-nya. Sistem *point of sale* akan dapat

melakukan login, menampilkan pesanan, mengatur user dan member, melakukan reservasi, menambahkan promo, mengganti status pesanan, dan menampilkan laporan penjualan dan stok.

### **2.2.3 Penerapan Teknologi Responsive Web Design menggunakan Library Bootstrap Untuk Pembuatan Aplikasi E-Journal pada Yayasan Bina Darma Salatiga**

Penelitian ini berjudul “Penerapan Teknologi *Responsive Web Design* menggunakan Library Bootstrap Untuk Pembuatan Aplikasi E-journal pada Yayasan Bina Darma Salatiga” [7] oleh Hans Christian Wijayanto, Ramos Somya. Dalam penelitian ini dilakukan pengembangan terhadap sistem elektronik jurnal milik Yayasan Bina Darma Salatiga yang sudah ada dengan Teknologi *Responsive Web Design* dan *Library Bootstrap* supaya pengembangan sistem jurnal elektronik ini membantu pengelola jurnal dalam mengelola data jurnal dan dapat menampilkan jurnal dalam halaman *web* yang responsif.

## **2.3 Dasar Teori**

### **2.3.1 Point of sales**

*Point of sale* atau POS merupakan sistem yang menangani proses transaksi jual beli dan pelanggan melakukan pembayaran atas barang atau jasa kepada penjual, atau dengan kata lain *point of sale* adalah sistem kasir. Sistem POS modern lebih dari sekedar menangani transaksi pada umumnya, seperti penggunaan kartu kredit, menangani sistem persediaan, dan pelaporan penjualan. Sistem POS biasanya diintegrasikan dengan layanan lainnya untuk mempermudah proses manajemen bisnis.

Dengan menggunakan sistem POS ada beberapa kelebihan yang bisa didapatkan seperti berikut.

- a. Pelaporan terperinci untuk setiap penjualan yang dilakukan
- b. Perhitungan persediaan barang yang dapat dengan otomatis diperbarui pada tiap penjualan yang dilakukan
- c. Penyediaan data yang akurat dan mudah diakses

Dalam memilih sistem POS, ada beberapa faktor yang perlu diperhitungkan.

- a. Fitur Dasar – sistem POS seharusnya mudah untuk dilakukan pengaturan, mudah digunakan, dan memiliki antar muka yang *user friendly* sehingga mudah menggunakan sistem POS dan tidak memakan banyak waktu untuk memberikan pelatihan kepada kasir, serta dapat menghasilkan pelaporan penjualan yang mudah dipahami.
- b. *Adaptability and Evolution* – karena teknologi yang terus berkembang maka sistem POS juga harus secara reguler diperbarui dengan fitur baru sesuai perkembangan teknologi, seperti yang paling populer saat ini adalah sistem multi-platform, yang memungkinkan sistem POS dapat diakses di beberapa perangkat terutama di perangkat *mobile / smartphone* dan dapat diakses dimana saja dengan mudah.
- c. Pelaporan yang Kuat – tidak hanya memberikan laporan penjualan, namun pelaporan yang lebih mendetail, seperti *inventory*, penjualan per barang, penjualan harian, laporan yang dapat diekspor, atau diagram.
- d. Aksesibilitas Data – data harus dapat diakses dengan mudah kapan pun dan dimana pun, dan dapat diakses dengan mudah di berbagai platform dan perangkat, terutama di *smartphone*.
- e. Keamanan Data – data yang disimpan sistem POS harus terjamin keamanannya dan tidak menyimpan informasi data sensitif milik pelanggan. Dan data harus terjaga dari virus atau malware, maka penyimpanan data dapat dilakukan dengan menggunakan sistem cloud [8].

### 2.3.2 Web Responsif

*Web* responsif merupakan desain halaman *web* yang memungkinkan *web* tersebut diakses oleh semua pengguna dengan

perangkat yang berbeda dengan tampilan yang cocok, rapi, nyaman dilihat dan diakses dengan perangkat yang berbeda, biasanya PC, tablet, atau ponsel. Pada umumnya, halaman *web* tidak dapat menyesuaikan tampilannya dengan ukuran layar perangkat [9].

Dengan desain *web* responsif, halaman *website* akan memberikan tampilan yang berbeda sesuai ukuran layar dengan berubah ukuran secara dinamis untuk menyesuaikan layar perangkat yang mengakses. Kelebihan dari desain *web* responsif adalah pada pengguna *mobile* / ponsel yang juga menampilkan tampilan *website* yang nyaman digunakan untuk pengguna layar kecil.

Pengembangan *web* responsif dilakukan menggunakan CSS dan JavaScript untuk menyesuaikan layout halaman. Pada CSS biasanya sering digunakan media query yang memudahkan untuk mengubah gaya berdasarkan karakteristik dari perangkat yang merender materi, termasuk tipe tampilan, lebar, tinggi, orientasi dan bahkan resolusi. Media query yang paling sering digunakan dalam CSS adalah `min-width`, `max-width`, `min-height`, dan `max-height` untuk menyesuaikan ukuran tinggi dan lebar maksimum dan minimum [10].

Selain dengan menggunakan media query CSS, *Bootstrap* menyediakan sistem grid yang lebih mudah untuk mengatur tata layout *website* sesuai beberapa perangkat dengan ukuran layar berbeda. Prefix yang digunakan untuk mengatur grid halaman *website* menggunakan *Bootstrap* adalah `.col-xs-`, `.col-sm-`, `.col-md-`, dan `.col-lg-`, yang digunakan untuk mengatur kolom atau lebar dari halaman *web*, `xs` untuk ponsel dengan ukuran layar kurang dari 768px, `sm` untuk tablet dengan layar lebih dari 768px, `md` untuk laptop kecil dengan layar lebih dari 992px, `lg` untuk komputer atau laptop dengan layar lebih dari 1200px [11].

### 2.3.3 Bootstrap

*Bootstrap* merupakan *framework* font-end yang digunakan dalam pengembangan *web* yang lebih mudah dan lebih cepat. *Bootstrap*

menggunakan HTML dan CSS untuk template desain *web*, seperti typography, form, button, navigasi, dan komponen antarmuka lainnya, dan juga menggunakan ekstensi JavaScript. *Bootstrap* dikembangkan pada Twitter oleh Mark Otto dan Jacob Thornton, dan dirilis pada 2011 sebagai produk open-source di GitHub. *Bootstrap* bertujuan untuk memudahkan pengembangan *website* yang dinamis dan aplikasi *web*.

*Bootstrap* menyediakan banyak komponen antarmuka class yang dirancang bagi pengguna supaya dengan mudah mendesain tampilan *web* yang menarik. Selain antarmuka, *Bootstrap* juga menyediakan fitur grid yang membantu untuk mengatur layout halaman *web* dengan mudah dan cepat, serta membantu membuat desain halaman *web* yang responsif sesuai dengan ukuran layar perangkat dengan lebih mudah. Dengan *Bootstrap*, pengguna dapat dengan leluasa dalam mengembangkan tampilan *website* dan kesesuaian dengan layar yang berbeda, dan dapat mengubah tampilan *Bootstrap* sendiri dengan menggunakan CSS sendiri [12].

Sebelum menggunakan *Bootstrap* dalam penulisan kode di dalam HTML, perlu menuliskan beberapa syntax seperti berikut.

Jika menggunakan file *Bootstrap* yang telah diunduh dituliskan syntax berikut.

```
<link href="css/Bootstrap.min.css" rel="stylesheet">
```

Jika tidak ingin mendownload file *Bootstrap* dan memilih menggunakan *Bootstrap* secara online dituliskan syntax berikut [13].

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/Bootstrap/3.3.7
/css/Bootstrap.min.css">
```

Jika perlu menggunakan jQuery dituliskan syntax berikut.

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.1
1.3/jquery.min.js"></script>
```



Jika perlu menggunakan JavaScript dituliskan syntax berikut, jika file javascript tersedia offline dapat digunakan source sesuai folder JavaScript-nya [14].

```
<script  
src="https://maxcdn.bootstrapcdn.com/Bootstrap/3.3.7/  
js/Bootstrap.min.js"></script>
```

#### **2.3.4 Android Web Server**

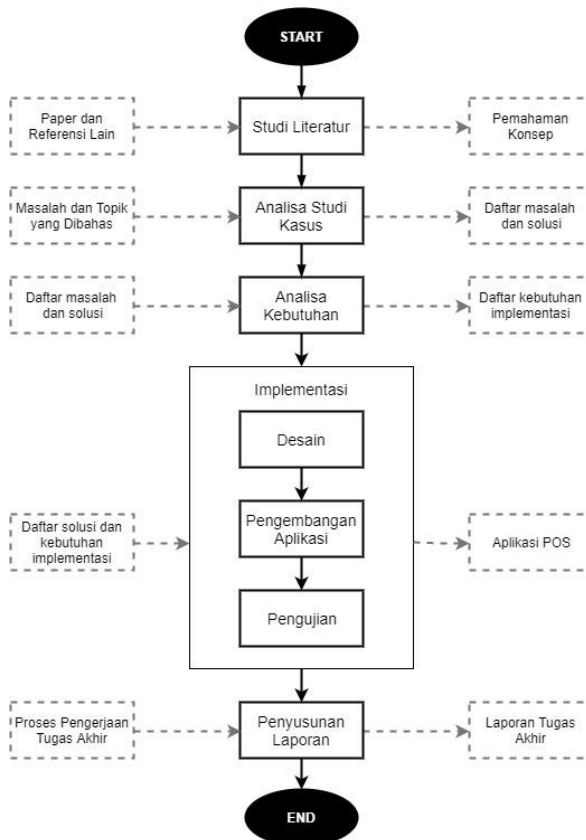
Android Web Server merupakan web server yang mendukung pengembangan web pada platform Android. Web server pada Android sama halnya dengan web server pada PC XAMPP, yang mendukung bahasa pemrograman PHP dan database MySQL. Ada banyak aplikasi penyedia web server untuk android yang dapat digunakan, dengan fitur yang kurang lebih sama. Pada umumnya aplikasi web server untuk Android sudah secara default berisikan file konfigurasi server, PHP, dan MySQL, yang sudah siap digunakan, sehingga tidak perlu melakukan pengaturan setelah instalasi aplikasi dan dapat diubah jika perlu diubah. Dan juga web server dapat dijalankan secara lokal pada smartphone pengguna dan tidak memerlukan berjalan dengan secara online / dengan sambungan internet. [16].

*Halaman ini sengaja dikosongkan*

## BAB III METODOLOGI PENELITIAN

### 3.1 Tahapan Tugas Akhir

Pada bagian ini akan dijelaskan alur dan metode dalam pengerjaan tugas akhir ini. Diagram metodologi penelitian ditampilkan dalam Gambar 3.1 berikut.



Gambar 3.1 Alur Pengerjaan Tugas Akhir

### **3.1.1 Studi Literatur**

Pada tahap ini dilakukan studi literatur terkait dengan topik penelitian, seperti *point of sale*, *framework Bootstrap*, perancangan *web* responsif, dan literatur lainnya yang berhubungan dengan masalah pengembangan aplikasi berbasis *web*. Beserta mengumpulkan literatur berupa paper penelitian terdahulu dengan topik yang serupa dengan topik yang sedang dikerjakan, seperti mengenai pengembangan aplikasi *point of sale*, dan pengembangan *web mobile* responsif untuk dijadikan sebagai sumber referensi untuk pengerjaan penelitian ini.

### **3.1.2 Analisa Studi Kasus**

Pada tahap ini dilakukan analisa terhadap studi kasus yang akan dijadikan sebagai topik penelitian, dimana disini untuk studi kasus menggunakan salah satu UMKM di Tulungagung yang merupakan toko oleh-oleh yang cukup besar, UD Intan Jaya. Disini dilakukan wawancara dengan pemilik UD Intan Jaya untuk mengetahui keseharian yang dilakukan oleh UD Intan Jaya dalam melakukan transaksi jual beli mereka, aktivitas membuka stand di bazar dan pameran, beserta cabang yang ada, juga masalah yang mungkin ada, terutama masalah yang berhubungan dengan *point of sale*, seperti transaksi di bagian kasir dan pengelolaan data penjualan.

### **3.1.3 Analisa Kebutuhan**

Setelah melakukan studi kasus dan analisa terhadap studi kasus, selanjutnya adalah melakukan analisa terhadap kebutuhan yang akan dilakukan pada penelitian ini, topik dari penelitian ini adalah pembuatan *point of sale* untuk UD Intan Jaya, maka dari itu kebutuhan yang akan digali adalah berkaitan dengan masalah yang dihadapi oleh UD Intan Jaya, data yang dibutuhkan dalam pembuatan aplikasi, preferensi pengguna, dan sebagainya.

### 3.1.4 Implementasi

#### 3.1.4.1 Desain

Selanjutnya, dari analisa kebutuhan yang didapat, dilakukan desain untuk pengembangan aplikasi. Melakukan desain terhadap database yang akan dibuat sesuai dengan data dan kebutuhan untuk membuat aplikasi *point of sale*, beserta desain antarmuka aplikasi yang akan dikembangkan.

#### 3.1.4.2 Pengembangan Aplikasi

Setelah pembuatan desain untuk database dan antarmuka aplikasi, pada tahap ini dilakukan pengembangan aplikasi *point of sale* yang berbasis *web* sesuai dengan solusi yang digagas dan kebutuhan, pengembangan aplikasi ini sesuai dengan desain antarmuka yang telah dirancang pada tahap sebelumnya dan desain database yang telah dibuat, pengembangan aplikasi menggunakan *framework Bootstrap* dengan sistem *grid*-nya yang lebih mudah untuk mengatur tata *layout website* sesuai beberapa perangkat dengan ukuran layar berbeda yang akan membantu dalam membuat antarmuka aplikasi *web* bisa responsif dan cocok untuk diakses melalui PC ataupun perangkat *mobile smartphone* atau *tablet*.

#### 3.1.4.3 Pengujian

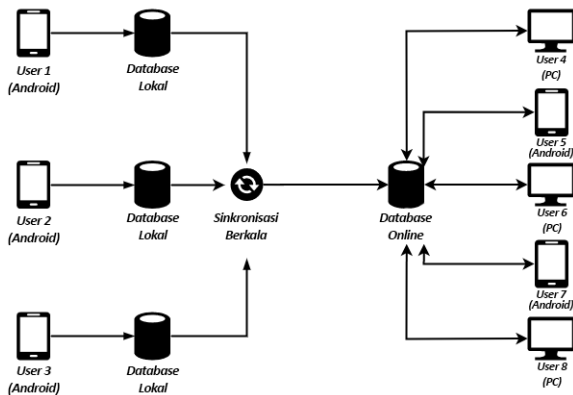
Dari aplikasi yang telah dibuat dilakukan pengujian terhadap aplikasi. Pengujian aplikasi dilakukan dengan metode *Black Box Testing*. Pengujian dilakukan dengan melakukan eksekusi seluruh fungsi yang terdapat pada aplikasi untuk melihat apakah seluruh fungsi sudah berjalan dengan benar atau tidak. Diawali dengan mendaftar seluruh fungsi yang ada serta input dan output masing-masing, dilanjutkan dengan membuat skenario eksekusi untuk masing-masing fungsi serta input untuk masing-masing fungsi, termasuk input benar dan salah untuk dieksekusi, kemudian dibandingkan output yang dihasilkan dengan output yang seharusnya dihasilkan.

### 3.1.5 Penyusunan Laporan

Tahap terakhir pada tahapan penelitian ini adalah penyusunan laporan tugas akhir, walaupun pengerjaan laporan tugas akhir ini dilakukan bersamaan dengan aktivitas lainnya. Pada laporan tugas akhir ini berisikan seluruh proses pengerjaan tugas akhir ini, beserta hasil akhir dan kesimpulan dari penelitian yang dilakukan.

### 3.2 Rancangan Arsitektur Sistem

Pada bagian ini akan digambarkan mengenai rancangan arsitektur dari sistem POS yang akan dibuat seperti pada Gambar 3.2 berikut.



Gambar 3.2 Rancangan Arsitektur Sistem POS

Pada Gambar 3.2 diatas digambarkan mengenai sistem *point of sale* akan berjalan secara *online* dan *offline* (lokal). Pada sistem POS lokal / *offline*, akan berjalan secara lokal pada masing-masing perangkat dengan terdapat *web server* lokalnya masing-masing pada tiap perangkat dan dengan basis datanya sendiri untuk tiap perangkat. Data pada basis data lokal nantinya adakan di-*backup* atau disinkronisasikan secara berkala entah itu tiap harinya, tiap minggu, atau tiap bulan ke basis data *online* untuk menyatukan



*Halaman ini sengaja dikosongkan*



## **BAB IV**

### **PERANCANGAN**

Pada bagian ini membahas mengenai alur perancangan terkait beberapa hal yang diperlukan dalam proses pembuatan aplikasi sesuai dengan alur yang dijelaskan pada Bab 3. Metode pengembangan yang dilakukan adalah dengan menggunakan metode ICONIX yang berdasarkan *use case* seperti RUP namun lebih kecil dan lebih ke dokumentasi kebutuhan dan desain lebih dari XP. Diagram UML yang digunakan untuk mendesain sistem aplikasi ini adalah diagram *use case*, *robustness*, *sequence*, dan, *class*.

#### **4.1 Requirement Review**

Pada tahap ini akan dilakukan analisa terhadap kebutuhan, untuk selanjutnya dibuat *domain model* dan *use case*.

##### **4.1.1 Analisa Kebutuhan**

Aplikasi dirancang sesuai dengan kebutuhan dari pengguna aplikasi ketika nanti diimplementasikan, yaitu pekerja dari UMKM UD Intan Jaya. Penggalan kebutuhan dilakukan dengan melakukan wawancara terhadap pemilik UMKM UD Intan Jaya mengenai aktivitas atau kegiatan yang biasa dilakukan untuk ditentukan kebutuhan yang dapat. Berikut adalah beberapa kebutuhan pengguna yang akan diterapkan ke dalam aplikasi.

- a. Dapat berjalan secara online dan offline
- b. Dapat berjalan di perangkat yang berbeda
- c. Antarmuka yang mudah dipahami

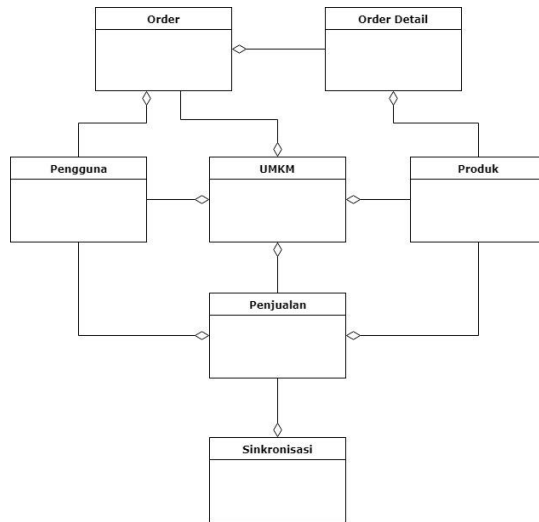
Berikut adalah kebutuhan fungsional aplikasi yang ditentukan berdasarkan pada penggalan kebutuhan dari pengguna sebelumnya.

- a. Pengguna (admin) dapat melakukan registrasi akun
- b. Pengguna (admin & pegawai) dapat melakukan login
- c. Pengguna (admin) dapat melakukan input produk
- d. Pengguna (admin) dapat melakukan edit produk

- e. Pengguna (admin) dapat menghapus produk
- f. Pengguna (admin) dapat menambah pengguna (pegawai)
- g. Sistem dapat menyimpan data produk
- h. Sistem dapat menampilkan produk
- i. Pengguna (admin) dapat melakukan sinkronisasi basis data lokal ke basis data remote
- j. Pengguna (admin & pegawai) dapat memilih dan memasukkan barang ke keranjang belanja beserta jumlahnya
- k. Pengguna (admin & pegawai) dapat menghapus barang dari keranjang belanja
- l. Sistem dapat menampilkan barang yang masuk ke keranjang belanja
- m. Sistem dapat menghitung dan menampilkan jumlah dan total harga barang
- n. Sistem dapat memperbarui stok barang sesuai transaksi yang dilakukan
- o. Pengguna dapat memasukkan jumlah uang yang dibayarkan
- p. Sistem dapat menghitung dan menampilkan kembalian uang yang dibayarkan
- q. Sistem dapat menampilkan struk belanja

#### **4.1.2 Domain Model**

*Domain model* berfungsi untuk menyamakan istilah yang akan pakai diproses selanjutnya, selama proses, *domain model* akan terus diperbarui menjadi lebih detail, hingga menjadi sebuah *class diagram*. Desain dari *domain model* awal ditampilkan pada Gambar 4.1.



Gambar 4.1 *Domain model* awal berdasarkan kebutuhan fungsional

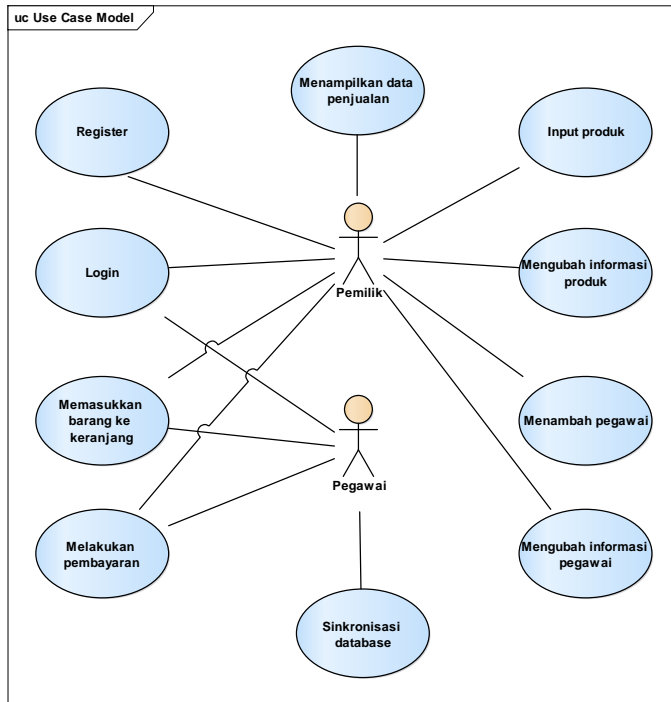
#### 4.1.3 Use Case

Disini akan dipetakan use case sesuai pada kebutuhan fungsional sistem yang telah didapat sebelumnya pada Tabel 4.1 beserta diagram use casenya pada Gambar 4.2. Dan berikut adalah aktor dalam sistem yang akan terlibat dalam penggunaan aplikasi nanti.

- Pemilik UMKM, merupakan pemilik UMKM dan orang yang mengelola akun utama dalam aplikasi ini yang dapat melakukan segala aktivitas di dalam sistem dan menambahkan pengguna baru yang akan berperan sebagai Pegawai.
- Pegawai, merupakan orang yang bertugas untuk melakukan transaksi dalam aplikasi ini, pegawai hanya dapat ditambahkan oleh pemilik UMKM, dan memiliki aktivitas yang terbatas di dalam sistem.

Tabel 4.1 Daftar *Use Case* dalam sistem

<b>UC ID</b>	<b>Use Case</b>
UC-01	Login
UC-02	Register
UC-03	Logout
UC-04	Input produk baru
UC-05	Edit informasi produk
UC-06	Menghapus produk
UC-07	Input pegawai baru
UC-08	Edit informasi pegawai
UC-09	Menghapus pegawai
UC-10	Menambah barang ke keranjang belanja
UC-11	Menghapus barang di keranjang belanja
UC-12	Menambahkan diskon
UC-13	Melakukan pembayaran
UC-14	Menampilkan data penjualan
UC-15	Melakukan sinkronisasi data penjualan



Gambar 4.2 Diagram *use case* dari sistem

#### 4.1.3.1 Use Case Description

*Use case description* untuk fungsi login ditunjukkan pada Tabel 4.2.

Tabel 4.2 *Use case description* fungsi login

UC-01 Login	
Ringkasan	Memasukkan akun yang telah dibuat agar pengguna dapat masuk ke dalam sistem
Aktor	Pemilik, Pegawai
Prasyarat	Pengguna sudah memiliki akun dan belum melakukan login

Kondisi Sukses	Pengguna masuk ke dalam aplikasi POS
Kondisi Gagal	Muncul pesan login gagal
Skenario Utama	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan username dan password yang sudah terdaftar</li> <li>2. Pengguna mengklik tombol login</li> <li>3. Pengguna masuk ke dalam aplikasi POS</li> </ol>
Skenario Alternatif	<ol style="list-style-type: none"> <li>1. Username dan / atau password yang dimasukkan salah atau belum terdaftar dan menampilkan pesan login gagal</li> </ol>

*Use case description* untuk fungsi registrasi ditunjukkan pada Tabel 4.3.

Tabel 4.3 *Use case description* fungsi Registrasi

<b>UC-02 Registrasi</b>	
Ringkasan	Pemilik UMKM mendaftarkan akunnya untuk dapat menggunakan aplikasi
Aktor	Pemilik,
Prasyarat	Pengguna belum memiliki akun
Kondisi Sukses	Akun pengguna yang juga mewakili UMKM berhasil didaftarkan
Kondisi Gagal	Muncul pemberitahuan registrasi gagal
Skenario Utama	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan data diri dan UMKM yang akan didaftarkan</li> <li>2. Pengguna mengklik tombol daftar</li> <li>3. Akun berhasil terdaftar di dalam aplikasi</li> </ol>
Skenario Alternatif	<ol style="list-style-type: none"> <li>1. Email atau username telah terdaftar dan menampilkan pemberitahuan registrasi gagal</li> <li>2. Dua password yang dimasukkan tidak sama dan menampilkan pemberitahuan registrasi gagal</li> </ol>

*Use case description* untuk fungsi *logout* ditunjukkan pada Tabel 4.4.

Tabel 4.4 *Use case description* fungsi *Logout*

<b>UC-03 Logout</b>	
Ringkasan	Pengguna keluar dari aplikasi
Aktor	Pemilik, Pegawai
Prasyarat	Pengguna sudah masuk ke dalam aplikasi
Kondisi Sukses	Pengguna keluar dari aplikasi
Skenario Utama	<ol style="list-style-type: none"> <li>1. Pengguna mengklik tombol logout</li> <li>2. Pengguna keluar dari aplikasi</li> </ol>

*Use case description* untuk fungsi input produk baru ditunjukkan pada Tabel 4.5.

Tabel 4.5 *Use case description* fungsi Input Produk Baru

<b>UC-04 Input Produk Baru</b>	
Ringkasan	Menambahkan produk baru untuk dimasukkan ke dalam aplikasi
Aktor	Pemilik
Prasyarat	
Kondisi Sukses	Produk berhasil ditambahkan ke aplikasi
Skenario Utama	<ol style="list-style-type: none"> <li>1. Pengguna mengklik tombol tambah produk</li> <li>2. Pengguna memasukkan informasi produk yang akan ditambahkan beserta foto / gambar.</li> <li>3. Pengguna mengklik tombol simpan</li> <li>4. Produk berhasil ditambahkan</li> </ol>

*Use case description* untuk fungsi edit produk ditunjukkan pada Tabel 4.6.

Tabel 4.6 *Use case description* fungsi Edit Produk

<b>UC-05 Edit Produk</b>	
Ringkasan	Mengedit informasi produk yang sudah ada
Aktor	Pemilik
Prasyarat	Produk sudah ada terdaftar
Kondisi Sukses	Informasi diubah sesuai informasi baru yang ditambahkan
Skenario Utama	<ol style="list-style-type: none"> <li>1. Pengguna mengklik tombol edit pada produk yang akan diedit</li> <li>2. Pengguna mengubah informasi yang ingin diubah</li> <li>3. Pengguna mengklik tombol simpan</li> <li>4. Informasi produk berubah sesuai informasi baru yang telah diubah</li> </ol>

*Use case description* untuk fungsi hapus produk ditunjukkan pada Tabel 4.7.

Tabel 4.7 *Use case description* fungsi Hapus Produk

<b>UC-06 Hapus Produk</b>	
Ringkasan	Menghapus produk yang sudah ada di aplikasi
Aktor	Pemilik
Prasyarat	Produk sudah terdaftar
Kondisi Sukses	Produk dihapus dari aplikasi
Skenario Utama	<ol style="list-style-type: none"> <li>1. Pengguna mengklik hapus pada produk yang ingin dihapus</li> <li>2. Pengguna memilih Iya</li> <li>3. Produk berhasil dihapus</li> </ol>
Skenario Alternatif	<ol style="list-style-type: none"> <li>1. Pengguna memilih Tidak</li> <li>2. Produk batal dihapus</li> </ol>

*Use case description* untuk fungsi menambah pegawai baru ditunjukkan pada Tabel 4.8.



Tabel 4.8 *Use case description* fungsi Menambah Pegawai Baru

<b>UC-07 Menambah Pegawai Baru</b>	
Ringkasan	Menambahkan pengguna baru sebagai pegawai
Aktor	Pemilik
Prasyarat	
Kondisi Sukses	Pegawai baru berhasil ditambahkan
Skenario Utama	<ol style="list-style-type: none"> <li>1. Pengguna mengklik tombol tambah pegawai</li> <li>2. Pengguna memasukkan data pegawai baru beserta fotonya.</li> <li>3. Pengguna mengklik simpan</li> <li>4. Pegawai baru berhasil ditambahkan</li> </ol>

*Use case description* untuk fungsi edit pegawai ditunjukkan pada Tabel 4.9.

Tabel 4.9 *Use case description* fungsi Edit Pegawai

<b>UC-08 Edit Pegawai</b>	
Ringkasan	Mengedit informasi pegawai yang sudah ada
Aktor	Pemilik
Prasyarat	Pegawai sudah ada terdaftar
Kondisi Sukses	Informasi diubah sesuai informasi baru yang ditambahkan
Skenario Utama	<ol style="list-style-type: none"> <li>1. Pengguna mengklik tombol edit pada pegawai yang akan diedit</li> <li>2. Pengguna mengubah informasi yang ingin diubah</li> <li>3. Pengguna mengklik tombol simpan</li> <li>4. Informasi pegawai berubah sesuai informasi baru yang telah diubah</li> </ol>

*Use case description* untuk fungsi hapus pegawai ditunjukkan pada Tabel 4.10.

Tabel 4.10 *Use case description* fungsi Hapus Pegawai

<b>UC-09 Hapus Pegawai</b>	
Ringkasan	Menghapus pegawai yang sudah ada di aplikasi
Aktor	Pemilik
Prasyarat	Pegawai sudah terdaftar
Kondisi Sukses	Pegawai dihapus dari aplikasi
Skenario Utama	<ol style="list-style-type: none"> <li>1. Pengguna mengklik hapus pada pegawai yang ingin dihapus</li> <li>2. Pengguna memilih Iya</li> <li>3. Pegawai berhasil dihapus</li> </ol>
Skenario Alternatif	<ol style="list-style-type: none"> <li>3. Pengguna memilih Tidak</li> <li>4. Pegawai batal dihapus</li> </ol>

*Use case description* untuk fungsi menambah barang ke keranjang belanja ditunjukkan pada Tabel 4.11.

Tabel 4.11 *Use case description* fungsi Memasukkan Barang ke Keranjang Belanja

<b>UC-10 Menambah Barang ke Keranjang Belanja</b>	
Ringkasan	Pengguna menambahkan barang ke daftar belanja sesuai produk yang tersedia ketika sedang transaksi
Aktor	Pemilik, Pegawai
Prasyarat	<ol style="list-style-type: none"> <li>1. Sudah terdaftar setidaknya satu produk</li> <li>2. Transaksi sebelumnya sudah selesai</li> </ol>
Kondisi Sukses	Barang yang dipilih masuk ke keranjang belanja
Skenario Utama	<ol style="list-style-type: none"> <li>1. Pengguna memilih barang pada aplikasi</li> <li>2. Pengguna memasukkan jumlah barang yang akan dimasukkan</li> </ol>

	<ol style="list-style-type: none"> <li>3. Pengguna mengklik OK</li> <li>4. Barang berhasil masuk ke dalam keranjang belanja</li> </ol>
--	--

*Use case description* untuk fungsi menghapus barang dari keranjang belanja ditunjukkan pada Tabel 4.12.

Tabel 4.12 *Use case description* fungsi Hapus Barang dari Keranjang Belanja

<b>UC-11 Menghapus Barang dari Keranjang Belanja</b>	
Ringkasan	Pengguna menghapus barang yang sudah ada pada keranjang belanja
Aktor	Pemilik, Pegawai
Prasyarat	Sudah ada setidaknya satu barang pada keranjang belanja
Kondisi Sukses	Barang yang dipilih berhasil dihapus dari keranjang belanja
Skenario Utama	<ol style="list-style-type: none"> <li>1. Pengguna mengklik icon hapus pada salah satu barang pada keranjang belanja</li> <li>2. Pengguna memilih Iya</li> <li>3. Barang yang dipilih berhasil dihapus dari keranjang belanja</li> </ol>
Skenario Alternatif	<ol style="list-style-type: none"> <li>1. Pengguna memilih Tidak</li> <li>2. Barang yang dipilih batal dihapus dari keranjang belanja</li> </ol>

*Use case description* untuk fungsi menambah diskon ditunjukkan pada Tabel 4.13.

Tabel 4.13 *Use case description* fungsi Menambah Diskon

<b>UC-12 Menambah Diskon</b>	
Ringkasan	Pengguna menambahkan diskon pada transaksi yang sedang berlangsung
Aktor	Pemilik, Pegawai

Prasyarat	Sudah ada setidaknya satu barang pada keranjang belanja
Kondisi Sukses	Diskon berhasil ditambahkan dan total harga berubah sesuai diskon yang dimasukkan
Skenario Utama	<ol style="list-style-type: none"> <li>1. Pengguna mengklik tombol tag</li> <li>2. Pengguna memilih satuan diskon, berupa Rupiah (RP) atau persen (%)</li> <li>3. Pengguna memasukkan jumlah diskon</li> <li>4. Pengguna mengklik OK</li> <li>5. Diskon berhasil ditambahkan dan total harga berubah sesuai dengan diskon</li> </ol>

*Use case description* untuk fungsi pembayaran ditunjukkan pada Tabel 4.14.

Tabel 4.14 *Use case description* fungsi Pembayaran

<b>UC-13 Melakukan Pembayaran</b>	
Ringkasan	Pengguna melakukan menyelesaikan transaksi yang sedang berlangsung dengan memasukkan nominal uang untuk melakukan pembayaran
Aktor	Pemilik, Pegawai
Prasyarat	Sudah ada setidaknya satu barang pada keranjang belanja
Kondisi Sukses	Nominal yang dimasukkan muncul di keranjang belanja beserta nilai kembalian
Skenario Utama	<ol style="list-style-type: none"> <li>1. Pengguna mengklik tombol bayar</li> <li>2. Pengguna memasukkan nominal uang yang dibayarkan sesuai customer</li> <li>3. Pengguna klik tombol OK</li> <li>4. Jumlah nominal dibayarkan muncul di keranjang belanja beserta nominal kembaliannya</li> </ol>

*Use case description* untuk fungsi menampilkan data penjualan ditunjukkan pada Tabel 4.15.

Tabel 4.15 *Use case description* fungsi Menampilkan Data Penjualan

<b>UC-14 Menampilkan Data Penjualan</b>	
Ringkasan	Pengguna menampilkan data penjualan yang telah dilakukan
Aktor	Pemilik
Prasyarat	Sudah ada setidaknya satu penjualan dilakukan
Kondisi Sukses	Data penjualan ditampilkan
Skenario Utama	<ol style="list-style-type: none"> <li>1. Pengguna mengklik menu Data Penjualan</li> <li>2. Klik detail pada salah satu baris data penjualan</li> <li>3. Detail data penjualan ditampilkan</li> </ol>

*Use case description* untuk fungsi sinkronisasi data penjualan ditunjukkan pada Tabel 4.16.

Tabel 4.16 *Use case description* fungsi Sinkronisasi Data

<b>UC-15 Melakukan Sinkronisasi Data Penjualan</b>	
Ringkasan	Pengguna mengunggah data penjualan yang telah dilakukannya sesuai akun masing-masing ke basis data <i>remote</i>
Aktor	Pemilik, Pegawai
Prasyarat	Sudah ada setidaknya satu data penjualan Data penjualan belum tersimpan di basis data <i>remote</i>
Kondisi Sukses	Data penjualan berhasil diunggah ke basis data <i>remote</i>
Kondisi Gagal	Data yang diunggah diabaikan dan tidak ada yang dimasukkan ke basis data <i>remote</i>

Skenario Utama	<ol style="list-style-type: none"> <li>4. Pengguna memilih jangka waktu transaksi penjualan yang akan diunggah</li> <li>5. Pengguna mengklik submit</li> <li>6. Data penjualan milik pengguna tersebut berhasil diunggah ke basis data remote</li> </ol>
Skenario Alternatif	<ol style="list-style-type: none"> <li>3. Data penjualan tidak ada atau sudah ada pada basis data <i>remote</i></li> <li>4. Data penjualan yang diunggah diabaikan dan tidak ada yang dimasukkan ke basis data <i>remote</i></li> </ol>

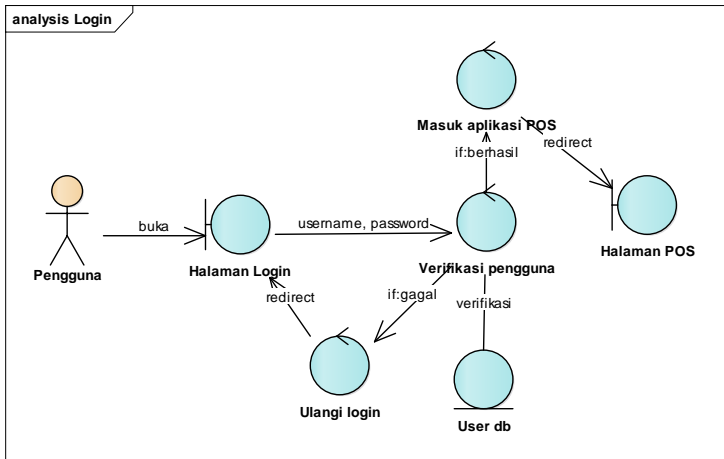
## 4.2 Preliminary Design Review

Setelah *use case* dibuat, selanjutnya menjelaskan bagaimana pengguna akan berinteraksi dengan sistem. Untuk menjelaskan hal tersebut maka disini dilakukan *robustness analysis* dan juga melakukan *update* pada *domain model*.

### 4.2.1 Robustness Analysis

Pada bagian ini akan dijelaskan mengenai diagram *robustness* dari sistem yang akan dibuat.

#### a. Login

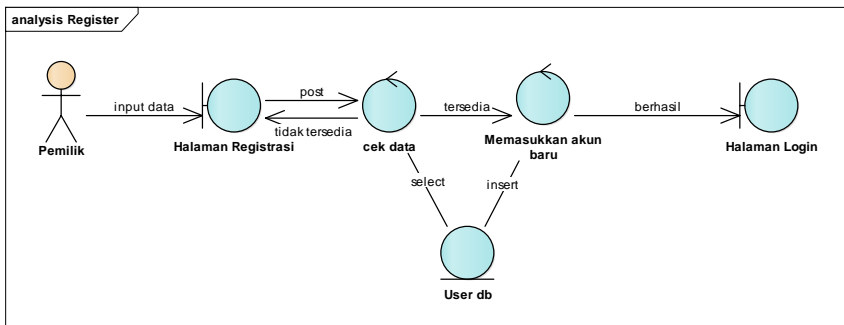


Gambar 4.3 *Robustness diagram* fungsi Login

Diagram *robustness* fungsi *login* ditunjukkan pada Gambar 4.1, untuk dapat masuk ke dalam aplikasi POS pengguna perlu *login* menggunakan *username* dan *password* pada halaman login yang kemudian akan diverifikasi ke basis data apakah akun yang dimasukkan sudah terdaftar, jika sudah maka akan diarahkan ke halaman POS, dan jika belum akan dikembalikan pada halaman *login* untuk mengulangi *login*.

b. Registrasi

Diagram *robustness* fungsi registrasi ditunjukkan pada Gambar 4.2, pengguna melakukan registrasi akun pada halaman registrasi dengan memasukkan data pengguna yang kemudian akan dimasukkan ke basis data untuk membuat akun baru, jika berhasil akan diarahkan ke halaman login untuk melakukan login, dan jika gagal akan dikembalikan pada halaman registrasi untuk mengisi ulang data yang didaftarkan.



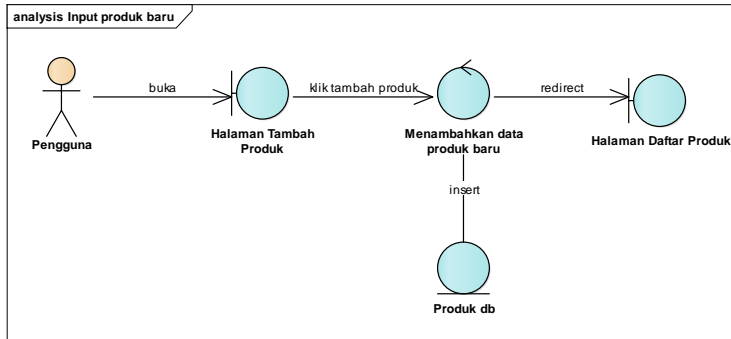
Gambar 4.4 *Robustness diagram* fungsi Registrasi

c. Input produk baru

Diagram *robustness* fungsi input produk baru ditunjukkan pada Gambar 4.3, pada halaman tambah produk pengguna mengisi informasi mengenai produk yang akan



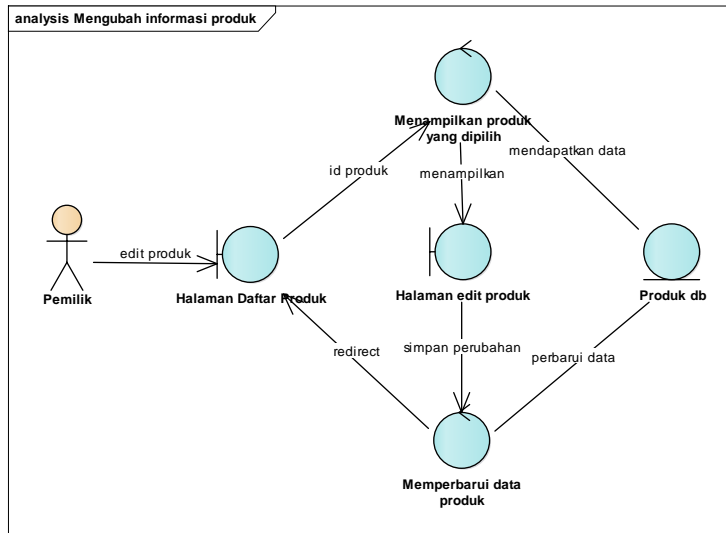
ditambahkan dan menyimpannya ke basis data, setelah selesai penambahan akan dialihkan ke halaman daftar produk.



Gambar 4.5 *Robustness diagram* fungsi Input Produk Baru

d. Edit informasi produk

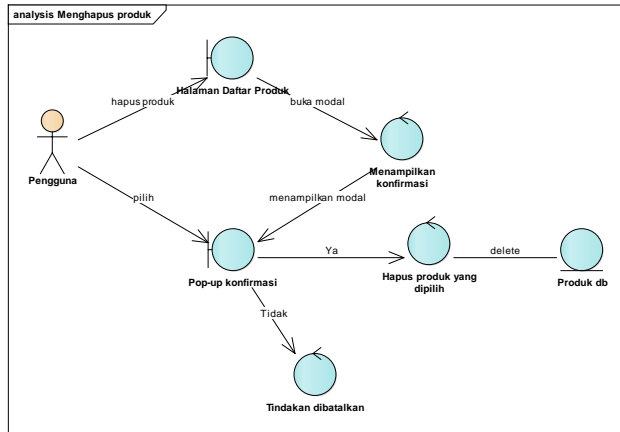
Diagram *robustness* fungsi *edit* informasi produk ditunjukkan pada Gambar 4.4, pengguna mengklik tombol edit pada salah satu produk di daftar produk, kemudian dialihkan ke halaman edit produk, pengguna mengubah informasi yang perlu diubah dan jika diklik simpan maka akan data pada basis data akan dipebarui, kemudian akan dialihkan kembali ke halaman daftar produk.



Gambar 4.6 *Robustness diagram* fungsi Edit Informasi Produk

e. Menghapus produk

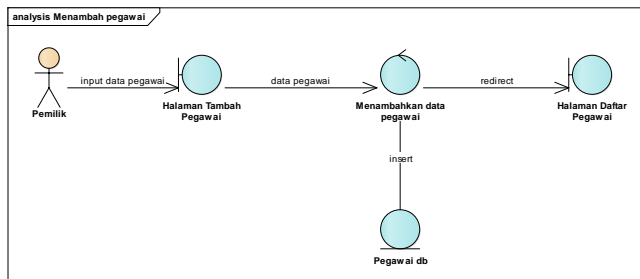
Diagram *robustness* fungsi menghapus produk ditunjukkan pada Gambar 4.5, pengguna mengklik hapus pada produk yang dipilih pada daftar produk, yang memunculkan *pop-up* konfirmasi, jika memilih iya produk akan dihapus, jika tidak maka akan dibatalkan dan *popup* ditutup.



Gambar 4.7 *Robustness diagram* fungsi Menghapus Produk

f. Input pegawai baru

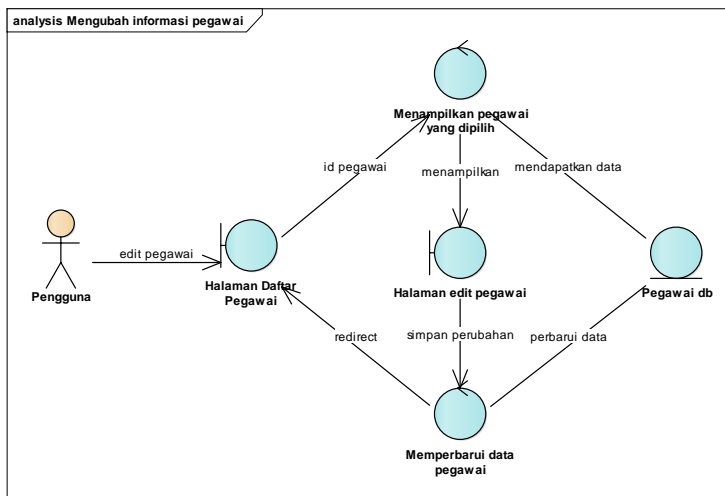
Diagram *robustness* fungsi input produk baru ditunjukkan pada Gambar 4.6, pada halaman tambah produk pengguna mengisikan informasi mengenai produk yang akan ditambahkan dan menyimpannya ke basis data, setelah selesai penambahan akan dialihkan ke halaman daftar produk.



Gambar 4.8 *Robustness diagram* fungsi Input Pegawai Baru

g. Edit informasi pegawai

Diagram *robustness* fungsi *edit* informasi pegawai ditunjukkan pada Gambar 4.7, pengguna mengklik tombol edit pada salah satu pegawai di daftar pegawai, kemudian dialihkan ke halaman edit pegawai, pengguna mengubah informasi yang perlu diubah dan jika diklik simpan maka akan data pada basis data akan dipebarui, kemudian akan dialihkan kembali ke



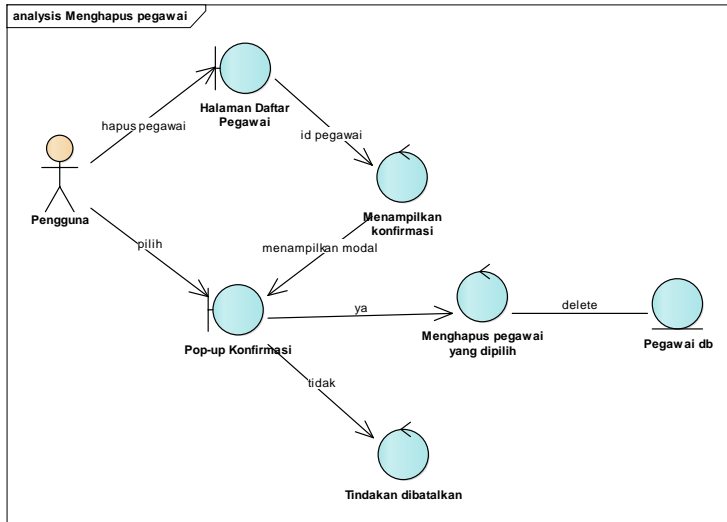
Gambar 4.9 *Robustness diagram* fungsi Edit Informasi Pegawai

halaman daftar pegawai.

h. Menghapus pegawai

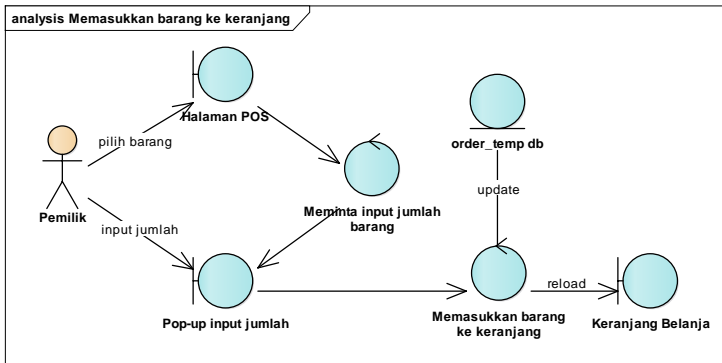
Diagram *robustness* fungsi menghapus pegawai ditunjukkan pada Gambar 4.8, pengguna mengklik hapus pada pegawai yang dipilih pada daftar pegawai, yang memunculkan *pop-up*

konfirmasi, jika memilih iya pegawai akan dihapus, jika tidak maka akan dibatalkan dan *popup* ditutup.



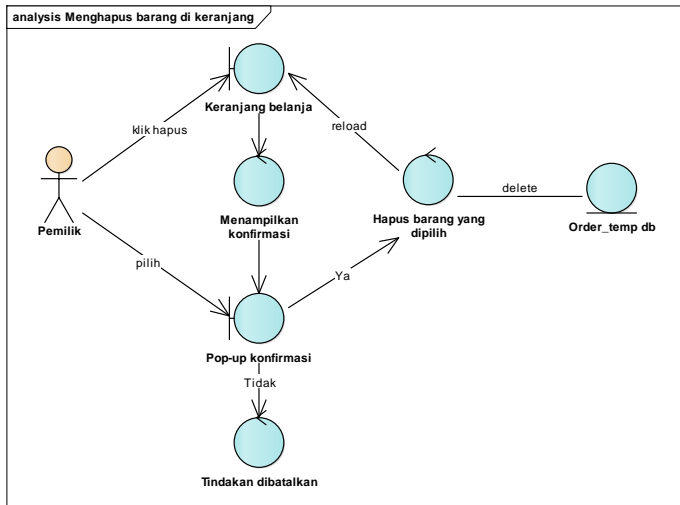
Gambar 4.10 *Robustness diagram* fungsi Menghapus Pegawai

- i. Memasukkan barang ke keranjang belanja  
 Diagram *robustness* fungsi memasukkan barang ke keranjang belanja ditunjukkan pada Gambar 4.9, pengguna memilih barang pada daftar produk dan memasukkan jumlah kemudian akan dimasukkan ke basis data dan ditampilkan pada keranjang belanja.



Gambar 4.11 *Robustness diagram* fungsi Menambahkan Barang ke Keranjang Belanja

- j. Menghapus barang di keranjang belanja  
 Diagram *robustness* fungsi menghapus barang dari keranjang belanja ditunjukkan pada Gambar 4.10, pengguna memilih b

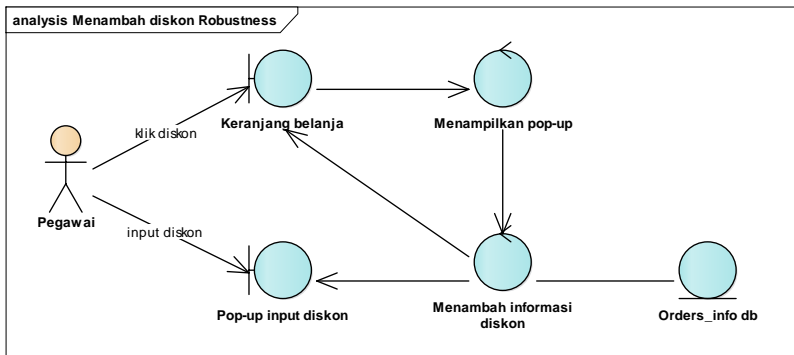


Gambar 4.12 *Robustness diagram* fungsi Menghapus Barang dari Keranjang Belanja

arang yang akan dihapus pada keranjang belanja, dan muncul *popup* konfirmasi, jika dipilih iya maka barang akan dihapus dari basis data dan keranjang belanja, jika tidak maka tindakan dibatalkan dan *popup* ditutup.

k. Menambahkan Diskon

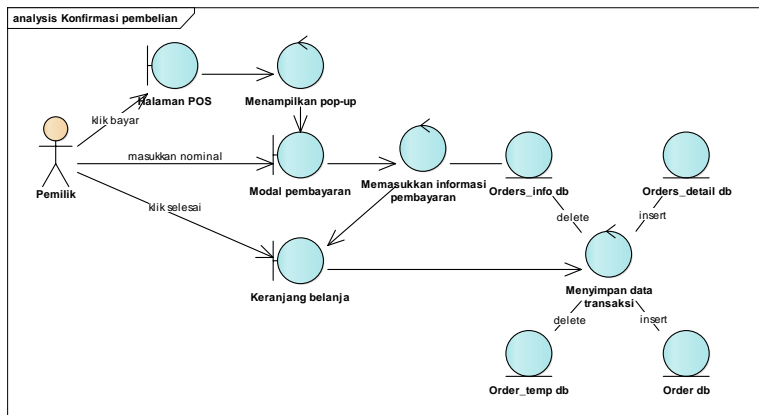
Diagram *robustness* fungsi menambahkan diskon ditunjukkan pada Gambar 4.11, pengguna mengklik tombol diskon pada keranjang belanja untuk menampilkan *pop-up* untuk memasukkan jenis diskon dan jumlah diskon, kemudian jenis dan jumlah diskon akan dimasukkan ke basis data tabel *orders\_info*, dan kemudian ditampilkan informasi diskon tersebut pada keranjang belanja.



Gambar 4.13 *Robustness diagram* fungsi tambah diskon

# 1. Pembayaran

Diagram *robustness* fungsi pembayaran ditunjukkan pada Gambar 4.12, pengguna mengklik tombol bayar pada keranjang belanja yang memunculkan *popup* pembayaran, dan memasukkan jumlah tunai dibayarkan. Jika selesai maka data penjualan akan dimasukkan pada basis data tabel *orders* dan *orders\_detail* dan menghapus pada *orders\_temp* dan *orders\_info*.

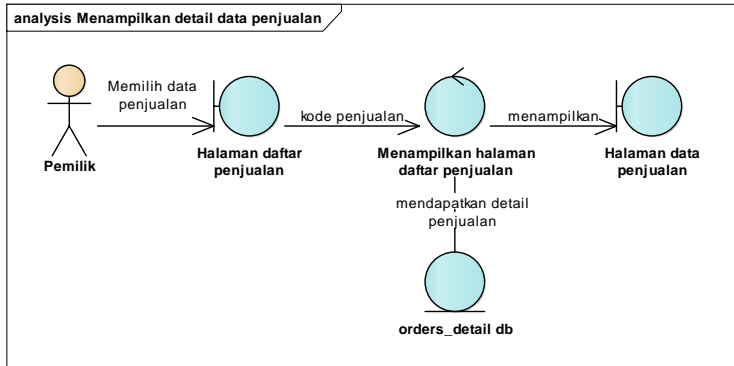


Gambar 4.14 *Robustness diagram* fungsi Menampilkan Data Penjualan

# m. Menampilkan data penjualan

Diagram *robustness* fungsi menampilkan data penjualan ditunjukkan pada Gambar 4.13, pengguna memilih *row* data penjualan yang akan dilihat detailnya, kode penjualan dari *row*



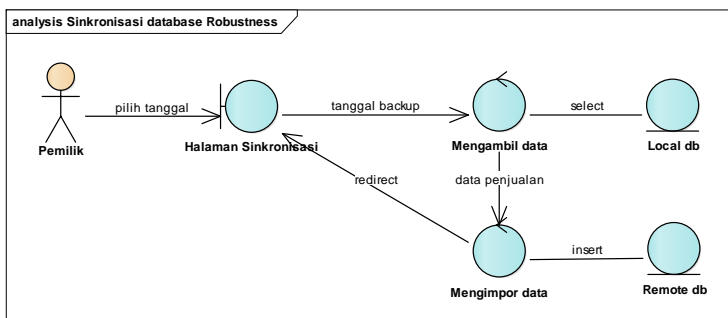


Gambar 4.16 Gambar 4.17 *Robustness diagram* fungsi Menampilkan Data Penjualan

yang dipilih akan di-*select* pada basis data kemudian ditampilkan detail penjualan dari kode penjualan yang dipilih.

#### n. Sinkronisasi data penjualan

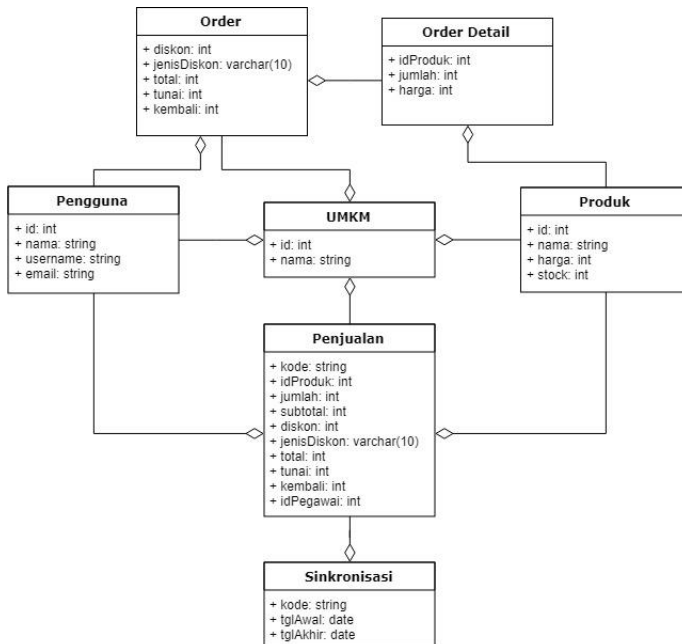
Diagram *robustness* fungsi sinkronisasi data penjualan ditunjukkan pada Gambar 4.14, pengguna akan memilih rentang waktu data yang akan disinkronisasikan, kemudian akan diambil data dari basis data sesuai rentang waktu yang dipilih untuk diimpor ke basis data *remote*.



Gambar 4.18 *Robustness diagram* fungsi Sinkronisasi Basis Data

#### 4.2.2 Updated Domain Model

*Domain model* selanjutnya akan diperbarui berdasarkan pada diagram *robustness* yang telah dibuat. *Domain model* diperbarui tidak hanya dengan memperbarui *class*-nya saja, namun juga mengisi atributnya. *Domain model* yang telah diperbarui ditunjukkan pada Gambar 4.19.



Gambar 4.19 *Domain model* dengan atribut

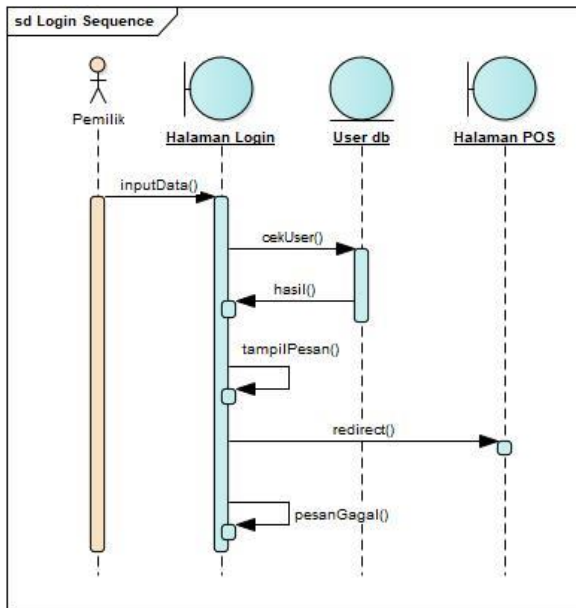
#### 4.3 Detailed Design Review

Selanjutnya dari diagram *robustness* yang telah dibuat dapat dibuat menjadi *sequence diagram* dan juga *domain model* dicek kembali dan diperbarui beserta melengkapinya dengan atribut dan operasi yang diperlukan untuk menjadi *class diagram*.

### 4.3.1 Sequence Diagram

#### a. Login

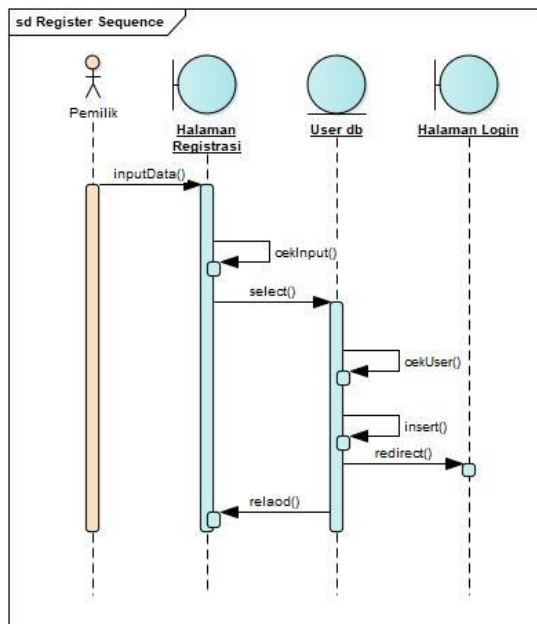
Diagram *sequence* dari fungsi *login* dapat dilihat pada Gambar 4.15, dimana pada halaman *login* pengguna memasukkan akun pengguna untuk masuk berupa *username* dan *password* dan kemudian divalidasi berdasarkan data pengguna yang sudah terdaftar pada basis data tabel *user*, jika data ditemukan maka *login* berhasil dan diarahkan masuk ke halaman *point of sale*, dan tidak ditemukan dan *login* gagal maka akan kembali ke halaman *login*.



Gambar 4.20 Sequence Diagram fungsi Login

b. Registrasi

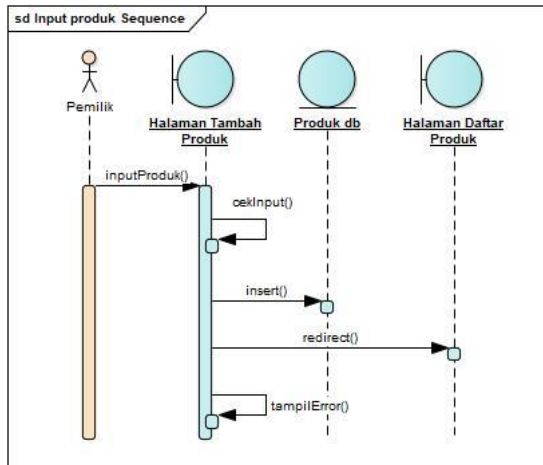
Diagram *sequence* dari fungsi registrasi dapat dilihat pada Gambar 4.16, pengguna memasukkan data pengguna untuk akun yang akan didaftarkan pada aplikasi, yang kemudian pada basis data dicek apakah ada akun dengan *email* atau *username* sudah digunakan, jika belum maka registrasi berhasil dan akun berhasil didaftarkan dan halaman dialihkan ke halaman *login*, dan jika sudah maka dikembalikan ke halaman registrasi.



Gambar 4.21 *Sequence Diagram* fungsi Registrasi

c. Input produk baru

Diagram *sequence* dari fungsi *input* produk baru dapat dilihat pada Gambar 4.17, pada halaman tambah produk, pengguna memasukkan informasi produk, seperti nama, harga, kategori, foto, dan, deskripsi, kemudian pengguna klik simpan, dan data akan disimpan pada basis data kemudian dialihkan ke halaman daftar produk dengan produk yang baru ditambahkan sudah berhasil ditambahkan.

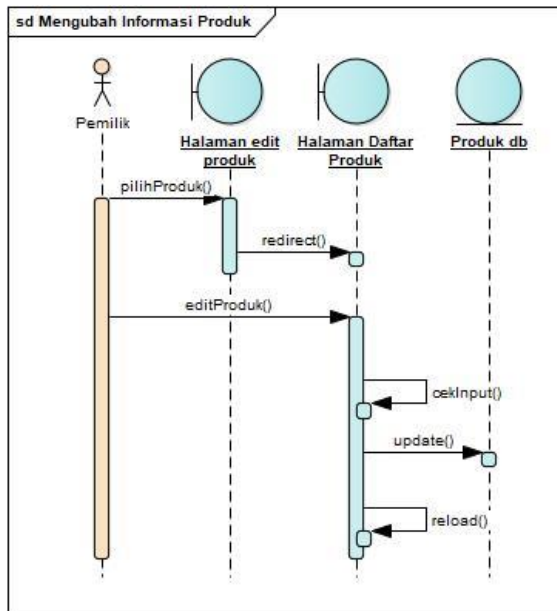


Gambar 4.22 *Sequence Diagram* fungsi Input Produk Baru

d. Edit informasi produk

Diagram *sequence* dari fungsi edit informasi produk dapat dilihat pada Gambar 4.18, pada daftar produk pengguna memilih produk yang ingin diedit dengan klik edit pada salah satu produk, kemudian dialihkan pada halaman edit produk, yang terdapat *form* yang sudah terisi dengan data produk, pengguna hanya perlu mengedit informasi yang perlu diubah

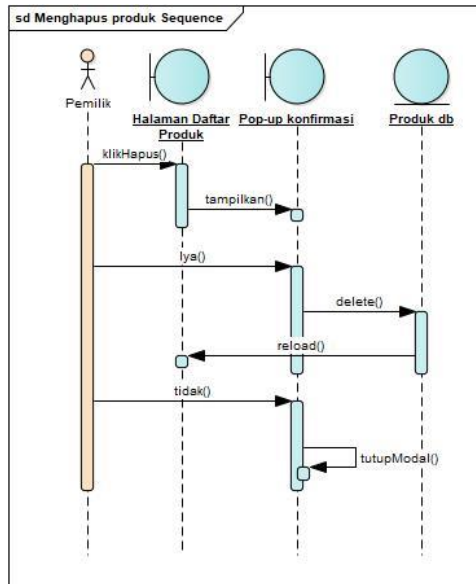
dan kemudian klik simpan, maka data pada basis data akan diedit sesuai dengan perubahan yang dilakukan.



Gambar 4.23 *Sequence Diagram* fungsi Edit Informasi Produk

e. Menghapus produk

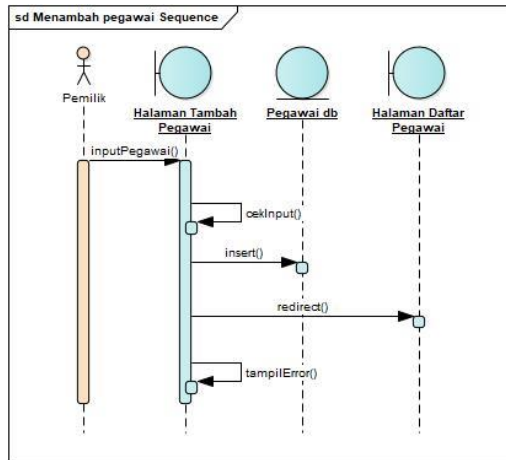
Diagram *sequence* dari fungsi hapus produk dapat dilihat pada Gambar 4.19, pada daftar produk pengguna dapat mengklik salah satu tombol hapus pada produk yang dikehendaki untuk dihapus, yang kemudian akan muncul *popup* untuk konfirmasi, jika iya maka produk akan dihapus dari basis data dan tidak muncul pada daftar produk, dan jika tidak maka tindakan akan dibatalkan dan *popup* ditutup.



Gambar 4.24 *Sequence Diagram* fungsi Menghapus Produk

f. Input pegawai baru

Diagram *sequence* dari fungsi *input* pegawai baru dapat dilihat pada Gambar 4.20, pada halaman tambah produk, pengguna memasukkan informasi pegawai, seperti nama, foto, dan *username* pegawai, kemudian pengguna klik simpan, dan data akan disimpan pada basis data kemudian dialihkan ke halaman daftar pegawai dengan akun pegawai yang baru ditambahkan sudah berhasil ditambahkan.

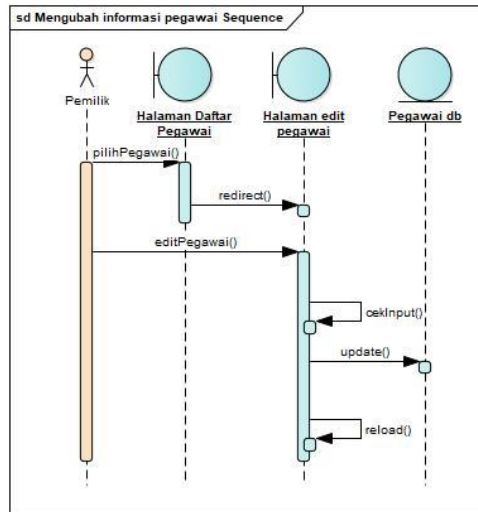


Gambar 4.25 Input Pegawai Baru

g. Edit informasi pegawai

Diagram *sequence* dari fungsi edit informasi pegawai dapat dilihat pada Gambar 4.21, pada daftar pegawai pengguna memilih pegawai yang ingin diedit dengan klik edit pada salah satu pegawai, kemudian dialihkan pada halaman edit pegawai, yang terdapat *form* yang sudah terisi dengan data pegawai, pengguna hanya perlu mengedit informasi yang perlu diubah dan kemudian klik simpan, maka data pada basis data akan diedit sesuai dengan perubahan yang dilakukan.

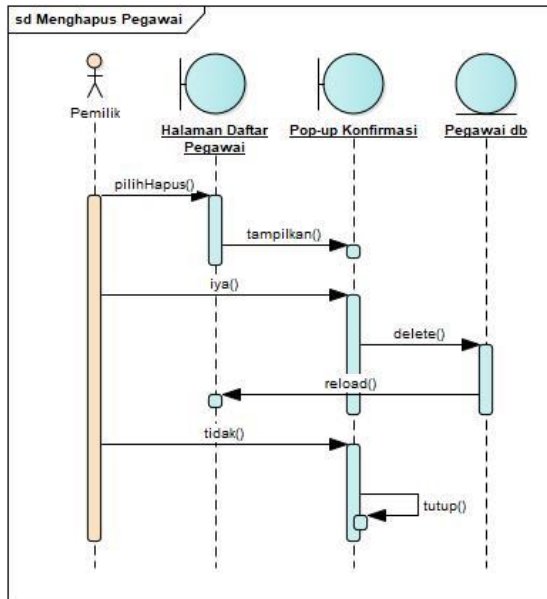




Gambar 4.26 *Sequence Diagram* fungsi Edit Informasi Pegawai

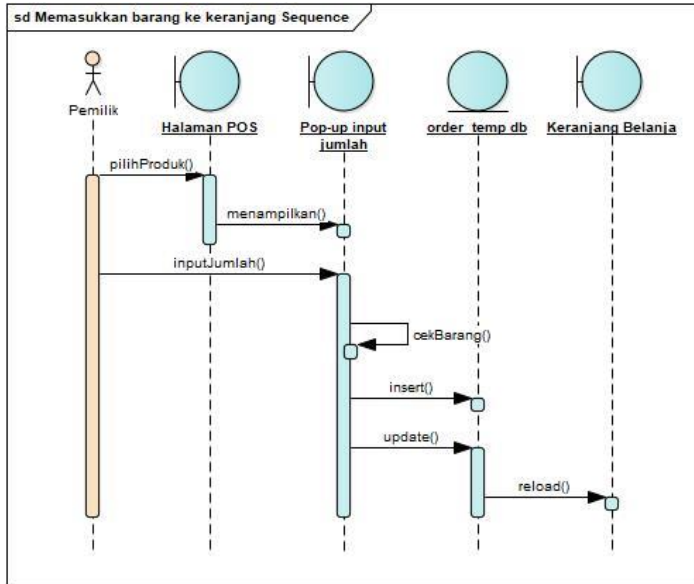
h. Menghapus pegawai

Diagram *sequence* dari fungsi hapus pegawai dapat dilihat pada Gambar 4.22, pada daftar pegawai pengguna dapat mengklik salah satu tombol hapus pada pegawai yang dikehendaki untuk dihapus, yang kemudian akan muncul *popup* untuk konfirmasi, jika iya maka pegawai dan akun pegawai akan dihapus dari basis data dan tidak muncul pada daftar pegawai, dan jika tidak maka tindakan akan dibatalkan dan *popup* ditutup.



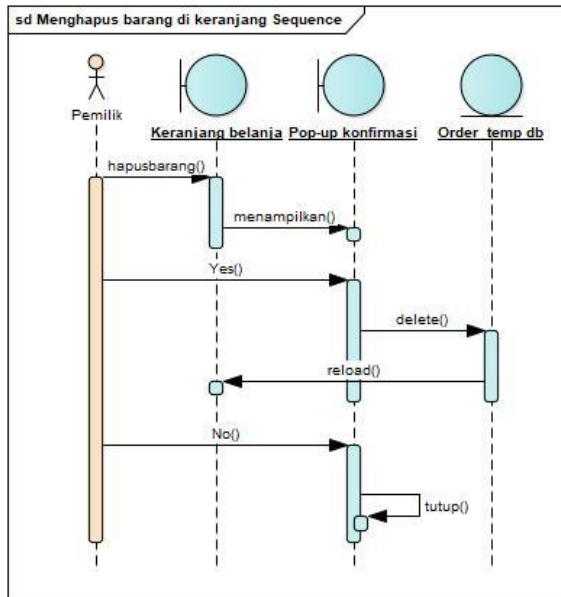
Gambar 4.27 *Sequence Diagram* fungsi Menghapus Pegawai

- i. Memasukkan barang ke keranjang belanja  
 Diagram *sequence* dari fungsi untuk memasukkan barang ke keranjang belanja dapat dilihat pada Gambar 4.23, pada halaman *point of sale*, pengguna mengklik salah satu produk yang akan dibeli oleh pembeli, kemudian akan memunculkan *popup* untuk memasukkan jumlah barang yang akan dibeli, selanjutnya klik OK maka produk dan jumlahnya akan dimasukkan ke basis data dan keranjang belanja akan dimuat ulang untuk menampilkan barang yang dibeli.



Gambar 4.28 *Sequence Diagram* fungsi Memasukkan Barang ke Keranjang Belanja

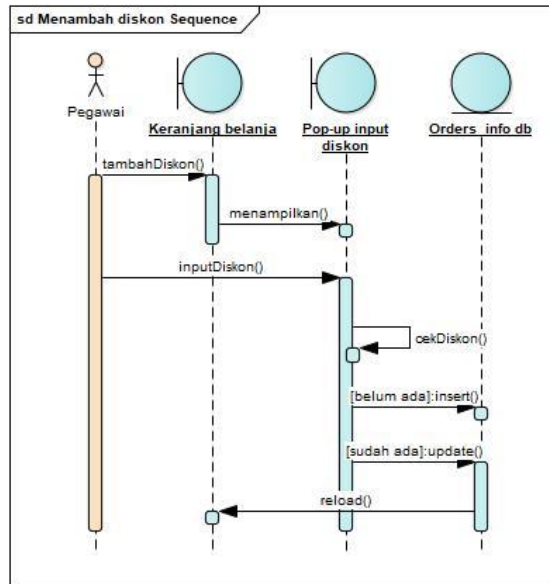
- j. Menghapus barang di keranjang belanja
- Diagram *sequence* dari fungsi untuk menghapus barang di keranjang belanja dapat dilihat pada Gambar 4.24, pada halaman *point of sale*, pengguna mengklik tombol hapus pada barang yang ingin dihapus pada keranjang belanja, saat diklik akan muncul *popup* untuk konfirmasi, jika dipilih iya, maka barang akan dihapus dari keranjang belanja juga basis data dan keranjang belanja akan dimuat ulang dan barang yang dihapus tidak akan ditampilkan lagi.



Gambar 4.29 *Sequence Diagram* fungsi Menghapus Barang dari Keranjang Belanja

k. Menambahkan diskon

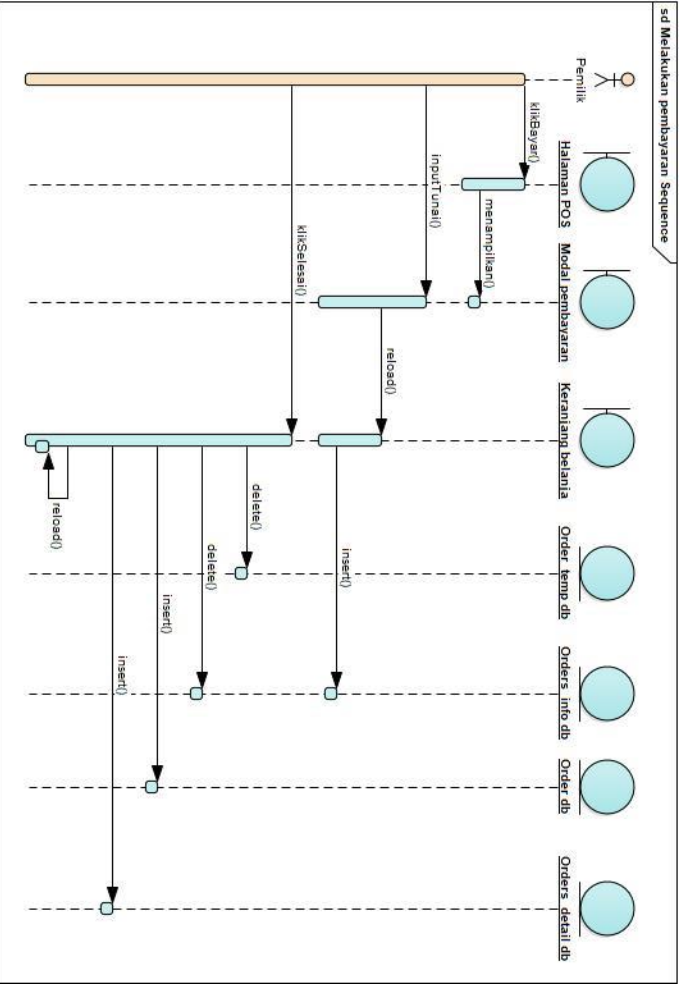
Diagram *sequence* dari fungsi diskon dapat dilihat pada Gambar 4.25, pada halaman *point of sale*, pengguna dapat mengklik tombol diskon pada bagian keranjang belanja yang akan memunculkan *popup* untuk memasukkan satuan apakah rupiah atau persen, dan jumlah diskonnya, jika sudah klik OK dan informasi mengenai diskon akan ditambahkan pada basis data dan keranjang belanja dimuat ulang dengan menampilkan informasi diskon yang ditambahkan.



Gambar 4.30 *Sequence Diagram* fungsi Tambah Diskon

#### 1. Melakukan pembayaran

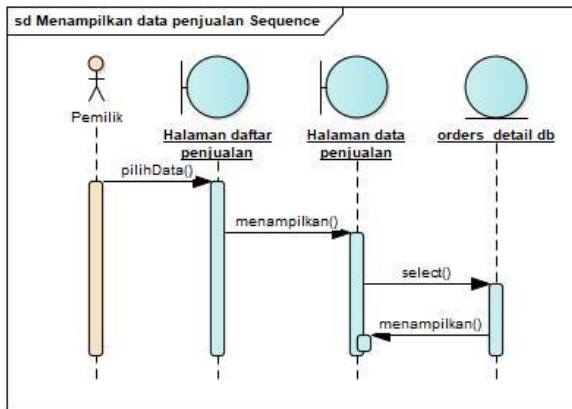
Diagram *sequence* dari fungsi pembayaran dapat dilihat pada Gambar 4.26, pada halaman *point of sale*, pengguna mengklik tombol bayar pada keranjang belanja, yang memunculkan *popup* untuk memasukkan jumlah tunai pembayaran yang dibayarkan oleh pembeli, kemudian klik OK, maka jumlah pembayaran akan dimasukkan ke basis data dan memperbarui data yang sudah ada, dan keranjang belanja akan dimuat ulang menampilkan jumlah pembayaran dan kembalian, dan mengganti tombol bayar dengan tombol edit untuk apabila jumlah yang dimasukkan salah, dan tombol selesai yang jika diklik akan memasukkan data final pada basis data yang sebelumnya masih sementara, dan menghapus keranjang belanja untuk dapat digunakan mulai awal lagi.



Gambar 4.31 *Sequence Diagram* fungsi Pembayaran

m. Menampilkan data penjualan

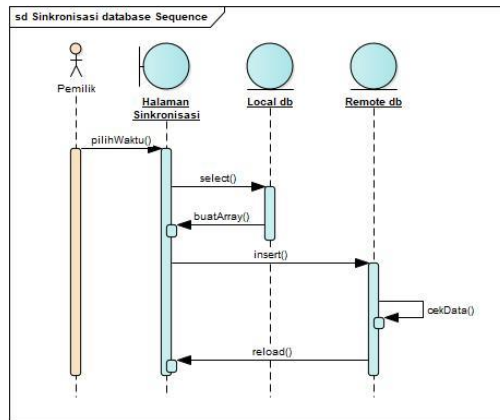
Diagram *sequence* dari fungsi menampilkan data penjualan dapat dilihat pada Gambar 4.27, pada halaman daftar penjualan pengguna dapat mengklik tombol detail pada salah satu *row* transaksi penjualan untuk dilihat data penjualannya, kemudian kode penjualan pada *row* yang dipilih akan di-*select* pada basis data untuk menampilkan detail transaksi sesuai kode penjualan, seperti pengguna yang melakukan transaksi, nama barang, jumlah, harga, subtotal, total, diskon, pembayaran, dan kembalian.



Gambar 4.32 *Sequence Diagram* fungsi menampilkan data penjualan

n. Sinkronisasi data penjualan

Diagram *sequence* dari fungsi sinkronisasi data penjualan dapat dilihat pada Gambar 4.28, pengguna akan memilih tanggal kapan data yang akan disinkronisasi pada halaman sinkronisasi, kemudian dari tanggal yang dipilih akan diambil data nya pada basis data dan kemudian data tersebut ditambahkan ke basis data yang berbeda (*remote*) lokasi data utamanya.

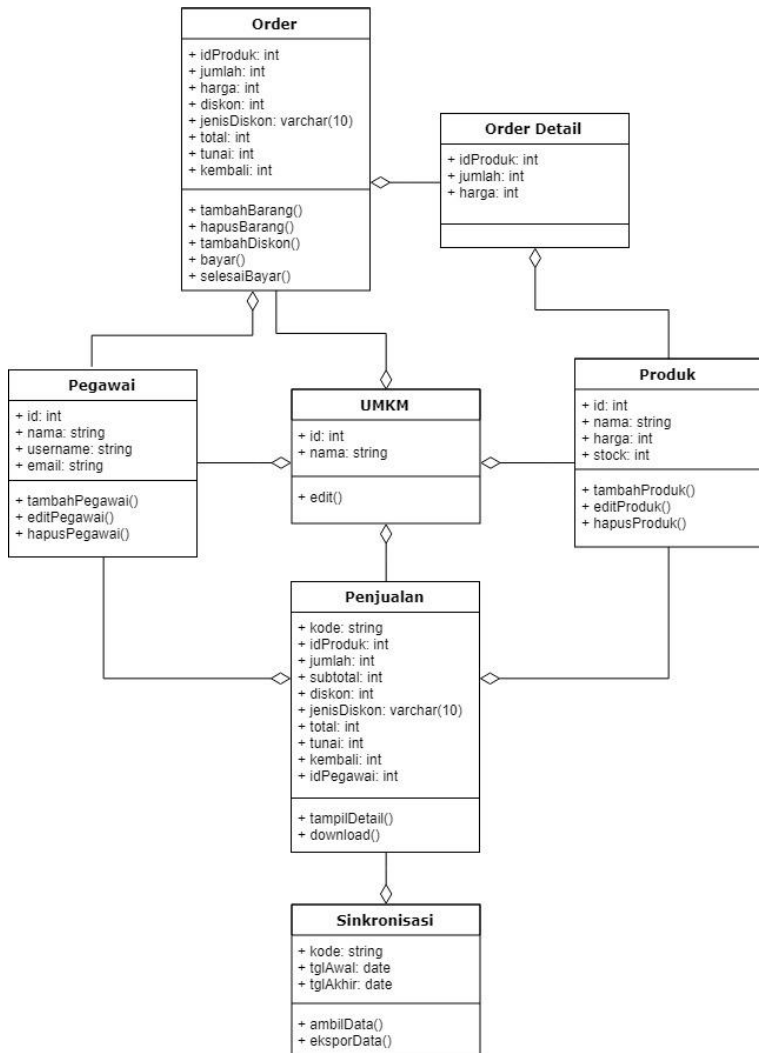


Gambar 4.33 *Sequence Diagram* fungsi Sinkronisasi Data

### 4.3.2 Class Diagram

*Domain model* kemudian dicek ulang kembali dan diperbarui, kali ini dilengkapi untuk atribut beserta operasinya dan menjadi sebuah *class diagram*. *Class diagram* yang dibuat dari *domain model* sebelumnya ditunjukkan pada Gambar 4.34.

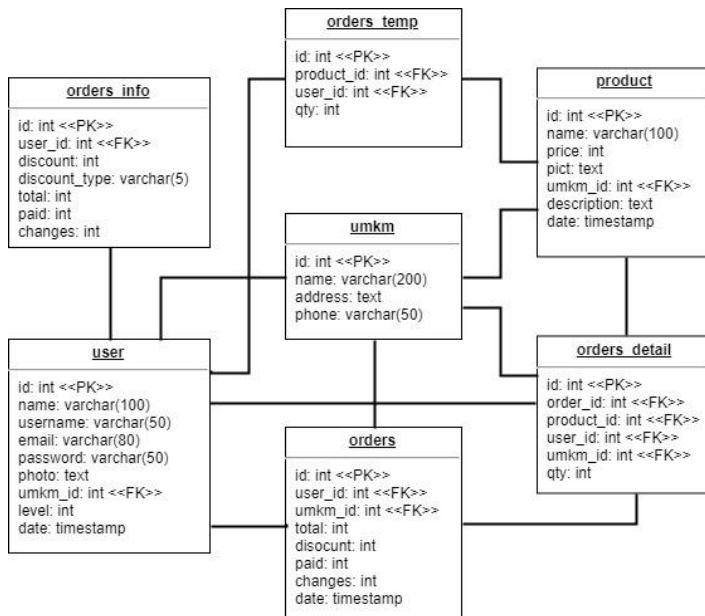




Gambar 4.34 Class diagram dari sistem

#### 4.4 Desain Basis Data

Desain basis data dari aplikasi *point of sale* ini dibuat berdasarkan pada kebutuhan pengguna yang telah didefinisikan sebelumnya beserta pada *class diagram* yang telah dibuat dan dikonversikan ke dalam skema basis data untuk diaplikasikan ke dalam aplikasi *point of sale* yang digambarkan pada Gambar 4.35



Gambar 4.35 Skema basis data aplikasi *point of sale*

Berikut adalah penjelasan mengenai tabel-tabel dalam skema basis data pada Gambar 4.35.

- user, berisi data mengenai pengguna yang akan menggunakan aplikasi, seperti email, password, username, dan id umkm.
- umkm, berisi informasi umkm, tabel ini dirujuk oleh tabel-tabel lainnya di basis data.

- c. `product`, berisi data mengenai produk yang dijual, seperti nama produk, harga, stok, dan gambar. Data produk dapat ditambah, diubah, dan dihapus oleh pengguna.
- d. `orders`, berisi data pembelian yang telah selesai dilakukan, hanya mencatat waktu pembelian dan pengguna yang melakukan proses pembelian. Untuk data lebih lanjutnya berada pada tabel `orders_detail`.
- e. `orders_detail`, berisi data lebih rinci dari pembelian pada tabel `orders`, seperti produk yang dibeli dan jumlahnya untuk tiap pembelian dari tabel `orders`.
- f. `orders_temp`, berisi data pembelian sementara yang dilakukan ketika aplikasi *point of sale* berjalan, pembelian yang dilakukan sementara akan masuk pada tabel ini dan akan dihapus setiap kali pembelian selesai dilakukan, dibatalkan, atau ketika halaman dimuat ulang.
- g. `orders_info`, berisi informasi sementara mengenai transaksi yang sedang dilakukan dan akan diselesaikan, seperti diskon, jumlah uang tunai yang dibayarkan, dan uang kembalian.


#### **4.5 Rancangan Desain Antarmuka**


Setelah mendapatkan kebutuhan dan membuat desain basis data, selanjutnya dibuat desain antarmuka yang akan diterapkan pada aplikasi web. Desain antarmuka ini dibuat sebagai acuan pengembangan aplikasi dan belum bersifat final karena dapat berubah ketika sedang dalam pengembangan. Berikut adalah beberapa modul aplikasi POS beserta dengan rancangan desain antarmuka masing-masing.

a. Halaman Login

Merupakan halaman awal dari aplikasi POS, halaman ini hanya berisikan form login untuk masuk ke aplikasi seperti pada Gambar 4.36, dan tombol register untuk mengarah ke halaman register jika belum memiliki akun.

**LOGIN**







— Belum punya akun? —


Gambar 4.36 Rancangan desain antarmuka halaman *Login*


b. Halaman Register

**DAFTAR**









— Sudah punya akun? —

Gambar 4.37 Rancangan desain antarmuka halaman Registrasi

Hampir sama dengan halaman login, hanya berisi form untuk daftar / register pengguna baru seperti ditunjukkan pada Gambar 4.37 dan tombol untuk mengarah ke halaman login jika sudah memiliki akun.

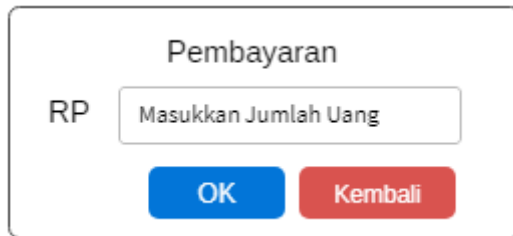
c. Halaman POS

Halaman utama setelah pengguna login dan masuk ke aplikasi seperti pada Gambar 4.38, merupakan halaman dimana kegiatan transaksi jual beli dilakukan, memilih barang, menghapus barang, dan melakukan pembayaran.

No	Barang	Jumlah	Sub-Total
1	Produk 1	2	RP 50.000
2	Produk 2	3	RP 150.000
3	Produk 3	1	RP 35.000
<b>TOTAL</b>			<b>RP 235.000</b>

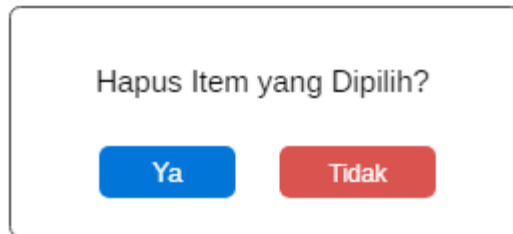
Gambar 4.38 Rancangan desain antarmuka halaman *Point of Sale*

Gambar 4.39 Rancangan desain antarmuka *modal po-up* memasukkan jumlah barang



A payment modal window with a white background and a thin gray border. At the top center is the title "Pembayaran" in a bold, dark gray font. Below the title, on the left, is the text "RP" in a bold, dark gray font. To the right of "RP" is a text input field with a light gray border and the placeholder text "Masukkan Jumlah Uang" in a dark gray font. Below the input field are two buttons: a blue button with the text "OK" in white, and a red button with the text "Kembali" in white.

Gambar 4.41 Rancangan desain antarmuka *modal pop-up* memasukkan jumlah pembayaran






A confirmation modal window with a white background and a thin gray border. At the top center is the title "Hapus Item yang Dipilih?" in a bold, dark gray font. Below the title are two buttons: a blue button with the text "Ya" in white, and a red button with the text "Tidak" in white.

Gambar 4.40 Rancangan desain antarmuka *modal pop-up* konfirmasi hapus barang

d. Halaman Daftar Produk

Halaman yang menampilkan data produk-produk dari UMKM, Gambar 4.42, selain menampilkan produk juga dapat melakukan tambah produk baru, edit produk dan hapus produk.

[+ Tambah Produk](#)

#	Gambar	Nama Produk	Harga	Stok	Action
1		Produk 1	RP 50.000	45	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; text-decoration: none;">Edit</a> <a href="#" style="background-color: #dc3545; color: white; padding: 2px 5px; text-decoration: none;">Hapus</a>
2		Produk 2	RP 35.000	56	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; text-decoration: none;">Edit</a> <a href="#" style="background-color: #dc3545; color: white; padding: 2px 5px; text-decoration: none;">Hapus</a>
3		Produk 3	RP 25.000	654	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; text-decoration: none;">Edit</a> <a href="#" style="background-color: #dc3545; color: white; padding: 2px 5px; text-decoration: none;">Hapus</a>

[<<](#)
[1](#)
[2](#)
[3](#)
[4](#)
[5](#)
[>>](#)

Gambar 4.42 Rancangan desain antarmuka halaman daftar produk

e. Halaman Tambah & Edit Produk

Diarahkan dari halaman daftar produk, halaman tambah produk dan edit produk memiliki tampilan yang sama, berisi form data produk, hanya pada edit produk, form sudah berisi data dari produk yang dipilih dari daftar produk.

### EDIT PRODUK

Nama Produk	<input type="text" value="Produk 1"/>
Harga	<input type="text" value="20.000"/>
Gambar	<input type="button" value="Choose File"/> No File Chosen
Stok	<input type="text" value="22"/>
Deskripsi	<div><div>Blablabla...</div><div></div></div>

KEMBALI

SIMPAN

Gambar 4.44 Rancangan desain antarmuka halaman Edit Informasi Produk

### INPUT PRODUK BARU

Nama Produk	<input type="text" value="Input"/>
Harga	<input type="text" value="Input"/>
Gambar	<input type="button" value="Choose File"/> No File Chosen
Stok	<input type="text" value="Input"/>
Deskripsi	<div><div></div><div></div></div>

KEMBALI

SIMPAN




Gambar 4.43 Rancangan desain antarmuka halaman Input Produk Baru



f. Halaman Daftar Pegawai

Halaman yang menampilkan data pegawai dari UMKM, selain menampilkan daftar pegawai juga dapat melakukan edit pegawai dan hapus pegawai, ditunjukkan pada Gambar 4.45.

[+ Tambah Pegawai](#)

#	Gambar	Nama Pegawai	Jenis Kelamin	Action
1		Pegawai 1	Laki-laki	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 3px;">Edit</a> <a href="#" style="background-color: #dc3545; color: white; padding: 2px 5px; border-radius: 3px;">Hapus</a>
2		Pegawai 2	Perempuan	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 3px;">Edit</a> <a href="#" style="background-color: #dc3545; color: white; padding: 2px 5px; border-radius: 3px;">Hapus</a>
3		Pegawai 3	Perempuan	<a href="#" style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 3px;">Edit</a> <a href="#" style="background-color: #dc3545; color: white; padding: 2px 5px; border-radius: 3px;">Hapus</a>

[«](#)

1

2

3

4

5

[»](#)

Gambar 4.45 Rancangan desain antarmuka halaman Daftar Pegawai

g. Halaman Tambah & Edit Pegawai

Diarahkan dari halaman daftar produk, halaman tambah produk, Gambar 4.46, dan edit produk, Gambar 4.47, memiliki tampilan yang sama, berisi form data produk, hanya pada edit produk, form sudah berisi data dari produk yang dipilih dari daftar produk.

### INPUT PEGAWAI BARU

Nama Pegawai	<input type="text" value="Input"/>
Jenis Kelamin	<input type="text" value="Input"/>
Foto	<input type="button" value="Choose File"/> No File Chosen
Alamat	<div><div></div><div></div></div>

Gambar 4.47 Rancangan desain antarmuka halaman Input Pegawai Baru

### EDIT PEGAWAI

Nama Pegawai	<input type="text" value="Pegawai 1"/>
Jenis Kelamin	<input type="text" value="Laki-laki"/>
Foto	<input type="button" value="Choose File"/> No File Chosen
Alamat	<div><div>Blablabla...</div><div></div></div>

Gambar 4.46 Rancangan desain antarmuka halaman Edit Informasi Pegawai

## **BAB V**

### **IMPLEMENTASI**

Pada bagian ini membahas mengenai implementasi dari kebutuhan fungsional yang telah didefinisikan pada bab sebelumnya ke dalam sistem yang diterapkan dalam bentuk kode.

#### **5.1.Lingkungan Implementasi**

Penegmbangan aplikasi ini menggunakan komputer dengan spesifikasi seperti pada Tabel 5.1, dan menggunakan tiga perangkat Android dengan spsifikasi seperti pada Tabel 5.2.

Tabel 5.1 Spesifikasi komputer yang digunakan

<b>Processor</b>	AMD A10-7400P Radeon R6
<b>Memory</b>	4.00 GB
<b>Sistem Operasi</b>	Windows 10 Pro 64-bit
<b>HDD</b>	931.50 GB

Tabel 5.2 Spesifikasi perangkat Android yang digunakan

<b>Sistem Operasi</b>	Android 5.0	Android 6.0	Android 7.0
<b>CPU</b>	Quad-core 1.3 GHz	Deca-core 2.11 GHz	Quad-core 1.4 GHz
<b>RAM</b>	2.00 GB	4.00 GB	3.00 GB
<b>Penyimpanan</b>	16.00 GB	64.00 GB	32.00 GB

Aplikasi dikembangkan dengan menggunakan beberapa perangkat lunak yang menunjang pengembangan aplikasi seperti pada Tabel 5.3.

Tabel 5.3 Teknologi yang digunakan

<b>Websserver</b>	XAMPP – Apache 2.4.28 KickWeb Server – DroidPHP for Android
<b>Text Editor</b>	Sublime Text 3.0
<b>Web Browser</b>	Google Chrome 67.0
<b>Database</b>	MariaDB 10.1.28 MySQL 5.1.62
<b>Library</b>	Bootstrap 3.3.7 jQuery 2.2.4

## 5.2. Konfigurasi Server

Pada bagian ini akan dijelaskan mengenai konfigurasi *web server* yang digunakan untuk menjalankan aplikasi pada perangkat PC dan *mobile* Android.

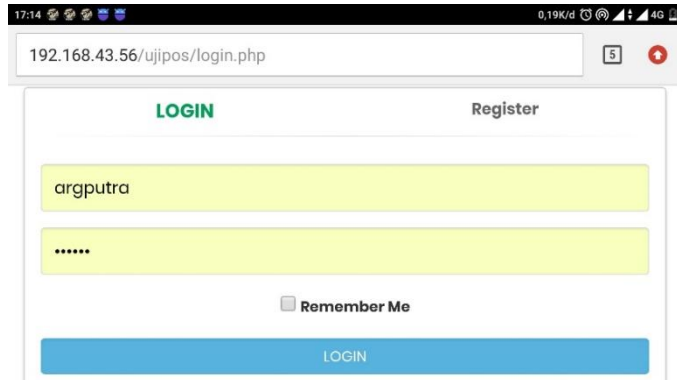
1. PC (Windows) – menggunakan XAMPP dengan Apache yang berjalan pada *port* 80 dan MySQL yang berjalan pada *port* 3306.
2. Mobile (Android) – menggunakan Kickwee Web Server dengan DroidPHP yang berjalan pada *port* 8080 dan MySQL yang berjalan pada *port* 3306.
3. *Remote / Online* – menggunakan penyedia hosting *web* gratis 000webhost

## 5.3. Pembuatan Aplikasi Point of Sale

Berikut akan dijelaskan mengenai pembuatan fungsi-fungsi pada aplikasi dengan penjelasan kode beserta tampilan fungsi pada halaman aplikasi.

### 5.3.1. Fungsi Login

Fungsi login ini merupakan bagian dimana pengguna, yaitu pemilik UMKM dan pegawai untuk masuk ke dalam aplikasi. Disini mereka harus memasukkan *username* dan *password* sesuai dengan yang telah didaftarkan di aplikasi. Tampilan dari halaman login dapat dilihat pada Gambar 5.1.



17:14 0.19K/d 4G

192.168.43.56/ujipos/login.php

LOGIN Register

argputra

.....

☐ Remember Me

LOGIN

Gambar 5.1 Tampilan halaman *Login*

Pengguna dapat masuk ke dalam aplikasi dengan memasukkan username dan password yang telah terdaftar pada form login, dan setelah itu klik pada tombol login untuk dilakukan verifikasi apakah username dan password yang dimasukkan sudah terdaftar dan benar, yang dieksekusi pada kode berikut ini.

```
1 <?php
2 if(isset($_POST["login"])) {
3     $sql = "SELECT * FROM user
4             WHERE username = '".$_POST["username"]."'
5             AND password = '".md5($_POST["password"])."'";
6     $result = mysqli_query($conn,$sql);
7     $user = mysqli_fetch_array($result);
8     if($user) {
9         $_SESSION["id"] = $user["id"];
10        if(isset($_POST["remember"])) {
11            setcookie ("member_login",$_POST["username"],time()+ (10
12                * 365 * 24 * 60 * 60));
13            setcookie ("password",$_POST["password"],time()+ (10 *
14                365 * 24 * 60 * 60));
15        }
16        else {
17            if(isset($_COOKIE["member_login"])) {
18                setcookie ("member_login","");
19            }
20            if(isset($_COOKIE["password"])) {
21                setcookie ("password","");
22            }
23        }
24        $message = "Login Gagal";
25    }
26 }
```

Gambar 5.2 Kode pemrograman fungsi Login

Ketika tombol login diklik, akan dijalankan query untuk mencari pengguna yang cocok dengan *username* dan *password* yang diinputkan oleh pengguna, jika pengguna ditemukan kemudian akan diset *session*-nya yaitu ID pengguna sesuai dengan ID yang sudah tercatat di basis data. Kemudian jika *checkbox Remember Me* dicentang oleh pengguna, maka akan diset *cookie username* dan *password* sesuai yang diinputkan tadi, jika tidak maka *cookie*-nya akan dikosongkan, dan jika *username* dan *password* yang

dimasukkan salah atau tidak ditemukan, maka akan memunculkan pesan login gagal.

Jika berhasil, setelah mengeset *session*, halaman akan dimuat ulang,

```
1 <?php
2 if(isset($_SESSION["id"])) { header(Location: "pos.php"); }
3
```

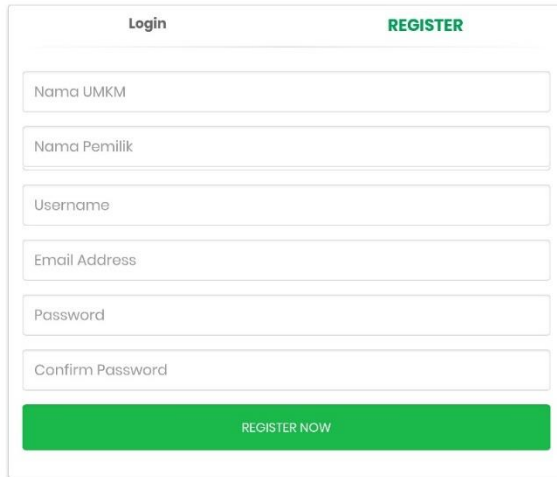
Gambar 5.3 Kode pemrograman jika pengguna sudah melakukan *login*

dan karena sudah melakukan *login* dan *session* sudah diset, seperti pada potongan kode pada Gambar 5.3, ketika membuka atau memuat ulang halaman *login* akan otomatis dialihkan ke halaman utama, yaitu halaman *point of sale*.

### 5.3.2. Fungsi Registrasi

Fungsi register merupakan dimana pengguna yang belum memiliki akun untuk masuk ke aplikasi dapat mendaftarkan diri dan UMKM-nya. Fungsi registrasi ini hanya digunakan untuk mendaftarkan sebagai UMKM, dengan kata lain hanya digunakan untuk mendaftar bagi pemilik UMKM, sedangkan akun untuk pegawai dapat didaftarkan oleh pemilik UMKM yang sudah memiliki akun. Tampilan dari halaman Register dapat dilihat pada Gambar 5.4.

## ARGOPOS



The registration form for ARGOPOS is displayed with the 'REGISTER' tab selected. It contains the following input fields: 'Nama UMKM', 'Nama Pemilik', 'Username', 'Email Address', 'Password', and 'Confirm Password'. A green 'REGISTER NOW' button is located at the bottom of the form.

Gambar 5.4 Tampilan halaman Registrasi

Pengguna harus mengisi seluruh inputan dengan benar untuk dapat mendaftar, dua kolom *password* harus diisi sama untuk menghindari ketidaksengajaan salah mengetik ketika memasukkan *password*. Jika semua kolom sudah diisi dengan benar kemudian klik Register untuk mendaftarkan akun. Setelah klik register akan dijalankan *query*, Gambar 5.5, untuk mencari adakah akun pada basis data dengan *username* atau *email* seperti yang dimasukkan oleh pengguna.

```
10 $sql = "SELECT * FROM user WHERE username = '$username' OR  
11 email = '$email' LIMIT 1";
```

Gambar 5.5 Kode *query select Username dan Email* pengguna



Selanjutnya, pada kode di Gambar 5.6, dicek, apakah *password* yang dimasukkan pada dua kolom *password* cocok atau tidak, jika tidak akan memunculkan pesan “Kedua password tidak sesuai”.

```

15     if ($password1 != $password2) {
16         $error_r = "Kedua password tidak sesuai";
17     }

```

Gambar 5.6 Kode verifikasi dua *password*

Kemudian pada Gambar 5.7, dicek apabila *username* atau *email* yang dimasukkan sudah ada akan dimunculkan pesan “Username sudah ada” atau “Email sudah ada”.

```

18     else if ($user) {
19         if ($user['username'] === $username) {
20             $error_r = "Username sudah ada";
21         }
22         if ($user['email'] === $email) {
23             $error_r = "Email sudah ada";
24         }
25     }

```

Gambar 5.7 Kode jika *username* atau *email* sudah ada

Jika semua pengecekan sudah dilewati dan tidak ada masalah, seperti pada potongan kode di Gambar 5.8, data yang diinputkan akan dimasukkan ke basis data, pertama akan memasukkan nama UMKM ke basis data, dengan memasukkan menggunakan `mysqli_insert_id()`, maka akan didapatkan id dari data yang baru saja dimasukkan, sehingga dapat ikut disertakan ketika selanjutnya memasukkan data pengguna, yaitu pemilik UMKM ke basis data sehingga data yang baru saja dimasukkan ke tabel UMKM dan tabel pengguna akan terhubung. Dan setelah data pengguna dimasukkan, akan memunculkan pesan “Registrasi Berhasil, Silahkan Login”, dan pengguna dapat melakukan *login* dengan menggunakan akun yang baru saja didaftarkan.

```

27     else {
28         $sql1 = "INSERT umkm (name)
29             : VALUES('$umkm')";
30         mysqli_query($conn, $sql1);
31         $umkm_id = mysqli_insert_id($conn);
32
33         $sql2 = "INSERT INTO user (name, email, username,
34             password, umkm_id, level)
35             : VALUES('".$name."', '".$email."', '".$username."
36             : ', '".md5($password1)."', '".$umkm_id."', '0')";
37         mysqli_query($conn, $sql2);
38         $message = "Registrasi Berhasil, Silahkan Login!";
39     }

```

Gambar 5.8 *Query* memasukkan data pengguna untuk didaftarkan

### 5.3.3. Fungsi Input Produk Baru

Fungsi ini adalah untuk menambahkan produk yang akan dijual dan ditampilkan pada aplikasi *point of sale*. Tampilan dari halaman input produk baru dapat dilihat pada Gambar 5.9. Pengguna memasukkan informasi mengenai produknya pada kolom yang tersedia beserta foto yang menggambarkan produk tersebut. Jika selesai, klik tombol Simpan, dan kemudian akan dieksekusi pada kode di Gambar 5.10, pertama akan dibaca *file* gambar yang diinputkan, kemudian mengganti nama mengubah nama *file* yang akan diupload ke direktori *website*, kemudian memasukkan *file* ke dalam direktori *assets/images/product/*.

Gambar 5.9 Tampilan halaman input produk baru

Selanjutnya informasi produk yang diinputkan pengguna pada *form input* produk tadi akan dimasukkan ke dalam basis data, dengan kode yang ditunjukkan pada Gambar 5.10, *file* gambar yang diupload tidak dimasukkan ke basis data, *file* gambar diupload pada direktori *website*, dan pada basis data hanya memasukkan nama *file* gambar yang diupload. Jika berhasil halaman akan

```

7      $random = substr(number_format(time() * rand(), 0, '', ''),
8          0, 4);
9      $images = $random . $_FILES['pic']['name'];
      $move = move_uploaded_file($_FILES['pic']['tmp_name'], '
          assets/images/product/' . $images);

```

Gambar 5.10 Kode untuk mendapatkan input *file* gambar

dialihkan ke halaman daftar produk. Penambahan produk baru hanya dapat dilakukan oleh pemilik UMKM.

### 5.3.4. Fungsi Edit Informasi Produk

Fungsi ini untuk mengubah atau mengganti informasi dari produk tertentu yang telah diinputkan sebelumnya dan sudah ada dalam basis data. Pengguna dapat mengedit informasi produk dari tombol yang ada di daftar produk, di setiap produk yang ada seperti pada

NO	NAMA PRODUK	GAMBAR	HARGA	KATEGORI	STOCK	DESKRIPSI	ACTION
1	Sosis Bakars		RP 67,000	Makanan	55	ZZZ	<div>Edit</div> <div>Hapus</div>

Gambar 5.12 Lokasi tombol Edit pada daftar produk

Gambar 5.11, yang kemudian halaman akan dialihkan ke halaman Edit Produk seperti pada Gambar 5.13. Yang dapat melakukan edit informasi produk hanya pemilik UMKM saja. Pada halaman edit produk terdapat form yang sudah terisi dengan informasi dari produk yang dipilih untuk diedit tadi, pengguna disini hanya tinggal mengganti beberapa bagian saja tanpa harus mengisi ulang seluruh kolom yang ada, dan kolom untuk *upload* gambar juga tidak perlu diisi lagi meski gambar yang sudah ada tidak ditampilkan. Jika data yang ingin diubah sudah selesai diubah pengguna dapat

```

11      $sqlinsert = "INSERT INTO product (name, price, pict,
12      category, stock, umkm_id, description)
13      VALUES ('" . $_POST['name'] . "',
14      '" . str_replace(".", "", $_POST['
15      price']) . "',
16      '" . $images . "',
17      '" . $_POST['category'] . "',
18      '" . $_POST['stock'] . "',
19      '" . $umkm_id . "',
20      '" . $_POST['description'] . "')";

```

data

menyimpan data dengan klik *Submit* yang kemudian akan diproses oleh kode pada Gambar 5.12.

## Edit Produk

Nama

Sosis Bakars

Gambar

Pilih File

Tidak ada file yang dipilih

Harga

67000

Kategori

Makanan

Deskripsi

ZZZ

Submit

Jika kolom gambar diinputkan file gambar baru, maka berarti gambar dari produk akan diubah, maka akan dibuat nama untuk file

Gambar 5.13 Tampilan halaman Edit Produk

gambar yang baru dan menentukan direktori penyimpanan file gambar dan kemudian dilakukan *query update* pada data produk di basis data dengan melakukan pembaruan dengan id sesuai id produk yang dipilih untuk diedit. Jika kolom gambar diabaikan dan tidak diinputkan file baru, maka akan diproses oleh kode pada Gambar 5.14.

```

1 <?php
2 if (!empty($_FILES) && $_FILES['pic']['error'] == 0) {
3     $random = substr(number_format(time() * rand(), 0, '', ''),
4         0, 4);
5     $images = $random . $_FILES['pic']['name'];
6     $move = move_uploaded_file($_FILES['pic']['tmp_name'], '
7         assets/images/product/' . $images);
8
9     $sqlinsert = "UPDATE product SET
10         name      = '" . $_POST['name'] . "',
11         price     = '" . str_replace(".", "", $_
12             POST['price']) . "',
13         pict      = '" . $images . "',
14         category  = '" . $_POST['category'] . "',
15         stock     = '" . $_POST['stock'] . "',
16         description = '" . $_POST['description'] . "
17         WHERE id   = '" . $_GET['id'] . "'";
18 }

```

Gambar 5.14 Kode untuk memperbarui data yang diedit jika gambar diinputkan

```

1 <?php
2 else {
3     $sqlinsert = "UPDATE product SET
4         name      = '" . $_POST['name'] . "',
5         price     = '" . str_replace(".", "", $_
6             POST['price']) . "',
7         category  = '" . $_POST['category'] . "',
8         stock     = '" . $_POST['stock'] . "',
9         description = '" . $_POST['description'] . "
10         WHERE id   = '" . $_GET['id'] . "'";
11 }

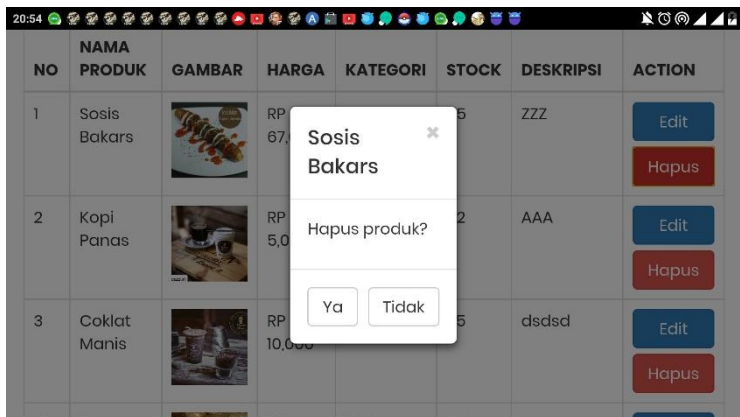
```

Gambar 5.15 Kode memperbarui data yang diedit tanpa input gambar

Jika kolom input *file* gambar baru dikosongi, maka kolom input gambar akan diabaikan, dan ketika melakukan *query update* ke basis data, kolom gambar diabaikan dan tidak dilakukan perubahan data.

### 5.3.5. Fungsi Menghapus Produk

Fungsi menghapus produk ini untuk menghapus produk yang sudah terdaftar sebelumnya, dan jika produk sudah tidak dijual lagi pengguna dapat menghapusnya dan tidak memenuhi daftar. Pengguna dapat menghapus produk dari tombol hapus yang ada pada daftar produk di sebelah dari tombol edit produk seperti pada Gambar 5.12. Ketika tombol hapus diklik, akan memunculkan



Gambar 5.16 *Pop-up* konfirmasi hapus

*modal* yang konfirmasi untuk menghapus produk, seperti pada Gambar 5.16. Jika pengguna memilih Iya maka akan diproses kode pada Gambar 5.17 untuk menghapus produk.

Setelah pengguna memilih Iya, maka akan dieksekusi *query* untuk menghapus produk pada basis data dengan id sesuai pada id produk yang dipilih. Jika pengguna memilih Tidak, maka *modal* akan ditutup dan hapus produk dibatalkan. Fungsi menghapus produk hanya dapat dilakukan oleh pemilik UMKM.

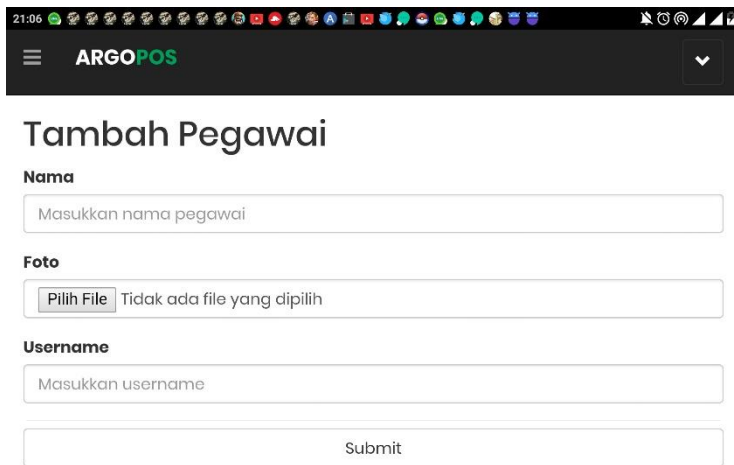
```

1  <?php
2  $sql = "DELETE FROM product
3          WHERE id = '$_GET['id']'";
4
5  mysqli_query($conn, $sql);
6
7

```

### 5.3.6. Fungsi Input Pegawai Baru

Fungsi ini adalah untuk menambahkan akun pegawai yang akan membantu pemilik UMKM menjalankan aplikasi dan melakukan transaksi. Tampilan dari halaman input pegawai baru dapat dilihat pada Gambar 5.18.



The screenshot shows a mobile application interface for 'ARGOPPOS'. The title of the page is 'Tambah Pegawai'. The form consists of the following elements:

- Nama**: A text input field with the placeholder text 'Masukkan nama pegawai'.
- Foto**: A file selection area with a 'Pilih File' button and the text 'Tidak ada file yang dipilih'.
- Username**: A text input field with the placeholder text 'Masukkan username'.
- Submit**: A large button at the bottom of the form.

Gambar 5.18 Tampilan halaman Tambah Pegawai

Pengguna memasukkan informasi mengenai pegawai yang akan ditambahkan pada kolom yang tersedia beserta foto pegawai, foto dapat dikosongi, namun akan diisi dengan foto sementara yang menggambarkan jika foto belum ada. Jika selesai, klik tombol Simpan, dan kemudian akan dieksekusi pada kode di Gambar 5.19, pertama akan dibaca *file* gambar yang diinputkan, kemudian mengganti nama mengubah nama *file* yang akan diupload ke direktori *website*, kemudian memasukkan *file* ke dalam direktori *assets/images/user/*.



```

1 <?php
2 $random = substr(number_format(time() * rand(), 0, '', ''), 0, 4);
3 $images = $random . $_FILES['pic']['name'];
4 $move = move_uploaded_file($_FILES['pic']['tmp_name'], 'assets/
    images/user/' . $images);
5

```

Gambar 5.19 Kode untuk mendapatkan input *file* gambar

Selanjutnya informasi pegawai baru yang diinputkan pada *form input* pegawai baru akan dimasukkan ke dalam basis data, dan pegawai diberikan *user level* 1, dimana pegawai hanya memiliki akses terbatas dibandingkan dengan pemilik UMKM dengan *user level* 0. Kode *query insert* informasi pegawai baru ditunjukkan pada Gambar 5.20. Setelah berhasil halaman akan dialihkan ke halaman daftar pegawai.

```

6 $sqlinsert = "INSERT INTO user (name, username, password, photo,
    umkm_id, level)
7             VALUES ('" . $_POST['name'] . "',
8             '" . $_POST['username'] . "',
9             '" . md5($_POST['password']) . "',
10            '" . $images . "',
11            '" . $umkm_id . "',
12            '1')";
13

```

Gambar 5.20 Kode *query* memasukkan data produk ke basis data

Penambahan pegawai baru hanya dapat dilakukan oleh pemilik UMKM.

### 5.3.7. Fungsi Edit Informasi Pegawai

Fungsi ini untuk mengubah atau mengganti informasi dari pegawai tertentu yang telah terdaftar sebelumnya dan sudah ada dalam basis data. Seperti pada fungsi edit produk, pengguna dapat mengedit informasi pegawai dari tombol edit yang ada di daftar pegawai, di setiap pegawai yang ada seperti pada Gambar 5.21, yang kemudian halaman akan dialihkan ke halaman Edit Pegawai seperti pada


Gambar 5.22. Yang dapat melakukan edit informasi pegawai hanya pemilik UMKM saja.






NO	FOTO	NAMA PEGAWAI	USERNAME	PASSWORD	ACTION
1		Pegawai 1	pegawai1	Sudah Diganti	<div>Edit</div> <div>Hapus</div>

Gambar 5.22 Lokasi tombol Edit pada daftar produk

Pada halaman edit pegawai terdapat form yang sudah terisi dengan informasi dari pegawai yang dipilih untuk diedit tadi, seperti pada fungsi edit produk, pengguna disini hanya tinggal mengganti beberapa bagian saja yang perlu diubah, termasuk kolom untuk *upload* gambar juga tidak perlu diisikan lagi meski gambar yang

21:32





### Edit Pegawai

**Nama**

Pegawai 1

**Foto**

Pilih File

Tidak ada file yang dipilih

**Username**

pegawai1

Submit

Gambar 5.21 Tampilan halaman Edit Pegawai

sudah ada tidak ditampilkan. Jika data yang ingin diubah sudah selesai diubah pengguna dapat menyimpan data dengan klik *Submit* yang kemudian akan diproses oleh kode pada Gambar 5.23.

```

1 <?php
2 if (!empty($_FILES) && $_FILES['pic']['error'] == 0) {
3     $random = substr(number_format(time() * rand(), 0, '', ''), 0, 4);
4     $images = $random . $_FILES['pic']['name'];
5     $move = move_uploaded_file($_FILES['pic']['tmp_name'], 'assets/
        images/product/' . $images);
6
7     $sqlinsert = "UPDATE user SET
8         name      = '" . $_POST['name'] . "',
9         photo     = '" . $images . "',
10        username  = '" . $_POST['username'] . "'
11        WHERE id   = '" . $id . "'";
12 }

```

Gambar 5.23 Kode untuk memperbarui data yang diedit jika gambar diinputkan

Jika kolom gambar diinputkan file gambar baru, maka berarti gambar dari produk akan diubah, maka akan dibuat nama untuk file gambar yang baru dan menentukan direktori penyimpanan file gambar dan kemudian dilakukan *query update* pada data produk di basis data dengan melakukan pembaruan dengan id sesuai id produk yang dipilih untuk diedit. Jika kolom gambar diabaikan dan tidak diinputkan file baru, maka akan diproses oleh kode pada Gambar 5.23.

```

1 <?php
2 else {
3     $sqlinsert = "UPDATE user SET
4         name      = '" . $_POST['name'] . "',
5         username  = '" . $_POST['username'] . "'
6         WHERE id   = '" . $id . "'";
7 }

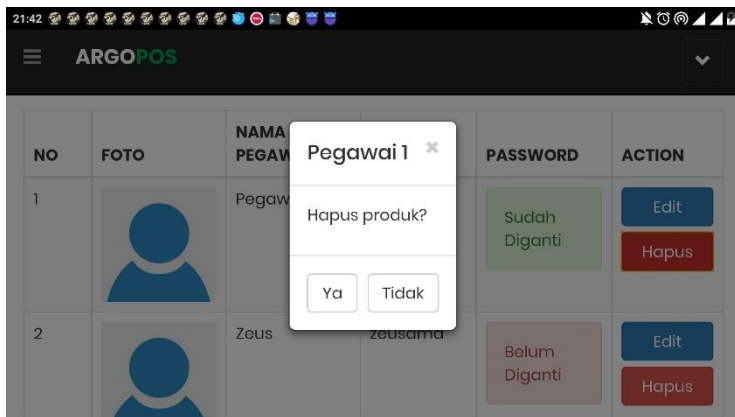
```

Gambar 5.24 Kode untuk memperbarui data yang edit tanpa input gambar

Jika kolom input *file* gambar baru dikosongi, maka kolom input gambar akan diabaikan, dan ketika melakukan *query update* ke basis data, kolom gambar diabaikan dan tidak dilakukan perubahan data.

### 5.3.8. Fungsi Menghapus Pegawai

Fungsi menghapus pegawai ini untuk menghapus pegawai yang sudah terdaftar sebelumnya. Seperti fungsi menghapus produk, pengguna dapat menghapus pegawai dari tombol hapus yang ada pada daftar pegawai di sebelah dari tombol edit produk seperti pada Gambar 5.25. Ketika tombol hapus diklik, akan memunculkan *modal* yang konfirmasi untuk menghapus produk, seperti pada Gambar 5.26. Jika pengguna memilih Iya maka akan diproses kode pada Gambar x untuk menghapus pegawai.



Gambar 5.25 *Pop-up* konfirmasi hapus

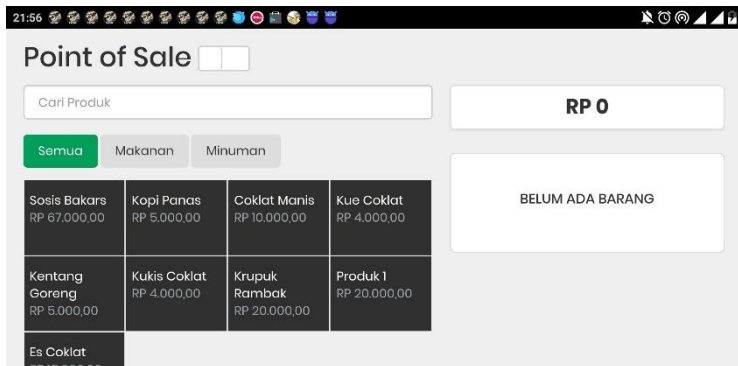
Setelah pengguna memilih Iya, maka akan dieksekusi *query* untuk menghapus pegawai pada basis data dengan id sesuai pada id pegawai yang dipilih. Jika pengguna memilih Tidak, maka *modal* akan ditutup dan hapus produk dibatalkan. Fungsi menghapus pegawai hanya dapat dilakukan oleh pemilik UMKM.

```
1 <?php
2 $sql = "DELETE FROM user
3       WHERE id = '$_GET['id']."'";
4
5 mysqli_query($conn, $sql);
```

Gambar 5.26 Kode *query* hapus pegawai yang dipilih

### 5.3.9. Point of Sale: Fungsi Menambahkan Barang ke Keranjang Belanja

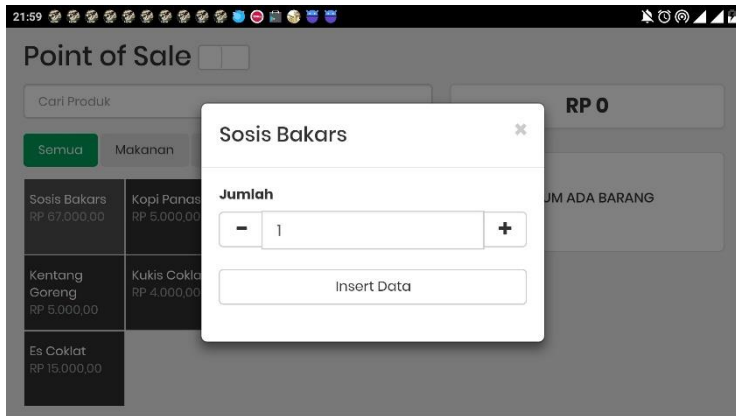
Fungsi pada *point of sale* ini adalah memilih produk / barang pada daftar yang berdasarkan pada produk yang sudah didaftarkan. Barang yang dipilih akan dimasukkan ke dalam keranjang belanja yang merupakan tempat transaksi berlangsung. Tampilan halaman *Point of Sale* ditunjukkan pada Gambar 5.27, dimana pada sebelah kiri adalah daftar barang yang dapat dipilih dan di sebelah kanan adalah keranjang belanja yang menunjukkan barang yang akan dibeli beserta dengan total harga pembelian. Dan jika pengguna memilih dan mengklik barang yang ada di daftar, maka akan menampilkan *modal* untuk memasukkan jumlah barang yang



Gambar 5.27 Tampilan halaman *Point of Sale*

dikehendaki seperti pada Gambar 5.28. Pada modal sudah disediakan *spinner* (tombol *plus minus*) bagi pengguna untuk mempermudah dan mempercepat dalam menginput jumlah barang, sehingga bagi pengguna *mobile*, tidak perlu menginputkan secara manual dan mengetikkan pada *virtual keyboard* pada perangkat *mobile*, yang terkadang mengganggu dan membutuhkan waktu untuk menunggu munculnya *keyboard* muncul di layar, apalagi jika perangkat sedang dalam keadaan melambat performa-nya. Fungsi

pada *spinner* dibuat menggunakan *library* dari *Javascript*, yaitu *jQuery* yang kode nya ditunjukkan pada Gambar 5.29.



Gambar 5.28 *Pop-up* input jumlah barang

Apabila pengguna mengklik tombol *plus* atau *minus*, maka akan mengambil *value* dari kolom dan diubah menjadi *integer*, kemudian *value* yang sudah diubah ke *integer* tadi akan ditambahkan 1 jika tombol *plus* yang diklik, dan dikurangi 1 jika tombol *minus* yang diklik, kemudian akan men-set *value* baru pada kolom input sesuai dengan *value* yang sudah diubah. Dan untuk tombol *minus*, jika *value* pada kolom adalah 1, maka jika dikurangi

```
1 $(document).on('click','plus',function() {
2     var x=parseInt($(".spinner-jml").val());
3     var y= x+1;
4     $(".spinner-jml").val(y);
5 });
6
7 $(document).on('click','min',function() {
8     var x=parseInt($(".spinner-jml").val());
9     var y= x-1;
10    if (y<1) {
11        y=1;
12    }
13    $(".spinner-jml").val(y);
14 });
```

Gambar 5.29 Kode fungsi *spinner*

lagi akan memunculkan angka 0 hingga seterusnya menjadi negatif, maka untuk mencegahnya, ditambahkan *if*, jika *value* < 1 maka akan di-set menjadi 1, sehingga *value* tidak akan mungkin bernilai kurang dari 1.

Setelah memasukkan jumlah barang yang diinginkan, kemudian klik OK, maka data akan diproses oleh kode pada Gambar 5.30.

```

1  $(document).on('click','.btn-cart',function() {
2      var prodid = $(this).attr('id');
3      var jumlah=$(("#jumlah"+prodid).val());
4      $.ajax({
5          type: "POST",
6          url: "cart.php",
7          data: {prodid:prodid,jumlah:jumlah},
8          dataType: "JSON",

```

Gambar 5.30 Kode fungsi tombol input jumlah

Ketika tombol OK pada modal diklik maka akan dijalankan fungsi pada *Javascript*, yang pertama akan mendefinisikan variabel id produk / barang yang dipilih, dan jumlah produk / barang yang dimasukkan. Kemudian kedua variabel tersebut akan di-*submit* - menggunakan AJAX ke URL *cart.php* yang berisi fungsi PHP untuk melakukan *insert* ke basis data secara POST dengan data id dan jumlah produk / barang yang dimasukkan. Untuk kode dari *cart.php* akan dijelaskan berikut ini. Pertama, pada Gambar 5.31, mendefinisikan variabel yang terdiri dari id pengguna, id UMKM,

```

1  <?php
2  session_start();
3  include 'userinfo.php';
4
5  $user_id = $_SESSION["id"];
6  $umkm_id = $det['umkm_id'];
7  $prodid = $_POST['prodid'];
8  $jumlah = $_POST['jumlah'];

```

Gambar 5.31 Kode definisi variabel yang digunakan

id produk, dan jumlah produk yang dimasukkan.

Kemudian, pada Gambar 5.32, melakukan *select* pada tabel *orders\_temp* untuk mengecek apakah barang yang dipilih oleh pengguna tersebut sudah ada pada keranjang belanja atau belum. Tabel *orders\_temp* berisi data keranjang belanja yang bersifat sementara yang akan dihapus ketika pembayaran sudah dilakukan.

```
24 $sql = "SELECT * FROM orders_temp
25     WHERE user_id = ".$user_id." AND product = ".$product."";
26 $result = mysqli_query($conn, $sql);
27
```

Gambar 5.32 Kode *query* untuk mencari barang yang dimasukkan

Jika barang oleh pengguna tersebut sudah ada, maka pada Gambar 5.33, data barang tersebut akan di-*update* dengan menambahkan jumlah barang yang sudah ada pada keranjang belanja dengan jumlah yang dimasukkan pengguna.

```
27
28 if (mysqli_num_rows($result) > 0) {
29     $sqlupdate = "UPDATE orders_temp
30         SET qty = qty+".$jumlah."
31         WHERE user_id = ".$user_id." AND product = ".$product."";
32     mysqli_query($conn, $sqlupdate);
33 }
```

Gambar 5.33 Kode *query* memperbarui data yang sudah ada

Jika barang belum ada, pada Gambar 5.34, maka barang yang akan dimasukkan tadi akan langsung dimasukkan ke dalam basis data dan keranjang belanja.

```
35 else {
36     $sqlinsert = "INSERT INTO orders_temp (product, umkm_id, user_id, qty)
37         VALUES ('".$product."', '".$umkm_id."', '".$user_id."', ".$jumlah."");
38     mysqli_query($conn, $sqlinsert);
39 }
```

Gambar 5.34 Kode *query* memasukkan data barang

Setelah selesai memperbarui data pada basis data, melanjutkan kode pada Gambar 5.35, pada Gambar 5.36, jika sukses maka akan dijalankan fungsi *reload()*, yang akan memuat ulang beberapa bagian (*div*) pada halaman *point of sale*. Bagian mana saja yang



perlu dimuat ulang dimasukkan ke dalam *array* yang dijadikan *parameter* pada fungsi *reload()*, selain fungsi *reload()*, juga menyembunyikan bagian yang memunculkan tulisan “Belum ada barang”. Dan jika gagal akan memunculkan *alert error*.

```

9      success: function(data) {
10          var obj = new Array (
11              "#cartList", "#cartList1", "#total", "#total1",
12              "#total-hid", "#cart-modal", "#mobile-footer",
13              "#modal-del", "#modal-dels", "#daftar-portrait"
14          );
15          reload(obj);
16          $("#cartEmpty").hide();
17      },
18      error: function(err) { alert(err); }
19  });
20  });

```

Gambar 5.35 Kode setelah dilakukan *post*

Fungsi *reload()* ditunjukkan pada Gambar x, yang mengakses *array* yang berisi id dan class dari *element* dari halaman *point of sale*, yang satu persatu akan dimuat ulang untuk memperbarui kontennya sehingga tidak perlu memuat ulang seluruh halaman

```

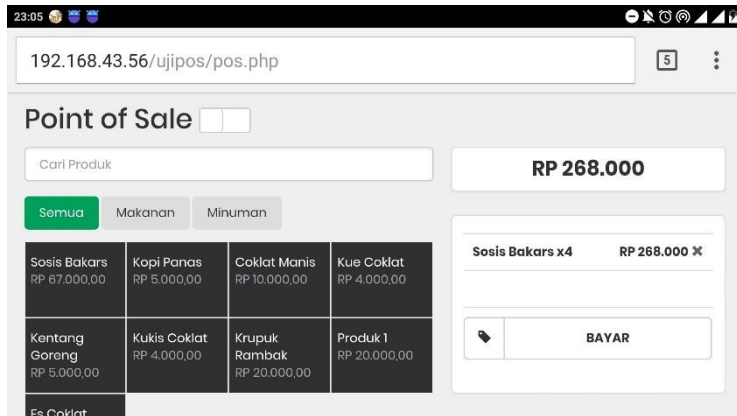
1  function reload(obj) {
2      $.each( obj, function(i, val ) {
3          $(val).load(location.href+" "+val+">*", "").hide().fadeIn('slow');
4      });
5  }

```

Gambar 5.36 Kode fungsi *reload*

web.

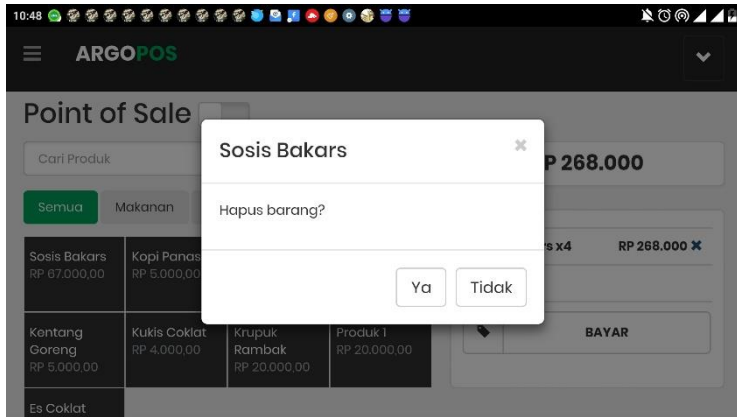
Dan untuk tampilan keranjang belanja setelah dimasukkan satu barang ditampilkan pada Gambar 5.37, disana ditampilkan nama barang, jumlah, dan subtotalnya, juga tombol silang (X) di sebelah barang untuk menghapus barang yang batal dibeli atau salah pilih, serta memunculkan tombol bayar dan diskon untuk melakukan pembayaran dan memberikan diskon. Panel total harga juga diperbarui sesuai dengan barang yang ada pada keranjang belanja.



Gambar 5.37 Tampilan keranjang belanja setelah dimasukkan produk

### 5.3.10. Point of Sale: Menghapus Barang dari Keranjang Belanja

Fungsi ini ada pada halaman *point of sale*, yaitu untuk menghapus barang yang sudah dimasukkan ke dalam keranjang belanja, jadi untuk menggunakan fungsi menghapus ini harus ada setidaknya satu barang pada keranjang belanja. Tombol hapus barang ada di samping harga subtotal barang pada keranjang belanja berupa tanda silang (X), lihat Gambar 5.37. Apabila pengguna mengklik tombol hapus, maka akan memunculkan *modal* konfirmasi apakah yakin menghapus barang tersebut, seperti pada Gambar 5.38.



Gambar 5.38 *Pop-up* konfirmasi hapus barang

Jika pengguna memilih Iya, maka akan dijalankan oleh kode pada Gambar 5.39 untuk menghapus barang, dengan fungsi *jQuery* mendapatkan id barang dari tombol hapus yang diklik untuk di-submit dengan AJAX ke *pos\_delete.php* dengan kode seperti pada Gambar 5.40.

```

1  $(document).on('click', '.btn-del', function() {
2      var prodid=$(this).attr('id');
3      $.ajax({
4          type: "POST",
5          url: "pos_delete.php",
6          data: {prodid:prodid},
7          dataType: "JSON",

```

Gambar 5.39 Kode tombol Iya (hapus)

Pada *pos\_delete.php* dijalankan *query* untuk menghapus produk sesuai dengan id produk yang dipilih untuk dihapus, produk yang dihapus adalah yang ada pada tabel *orders\_temp*.

```

1 <?php
2 $sql = "DELETE FROM orders_temp
3       WHERE product='".$$_POST['prodid']."'";
4
5 mysqli_query($conn, $sql);
6

```

Gambar 5.40 Kode *query* hapus barang

Setelah selesai dihapus, melanjutkan AJAX tadi, Gambar 5.41, jika sukses akan menjalankan fungsi *reload()* untuk memuat ulang beberapa bagian pada halaman untuk memperbarui kontennya.

```

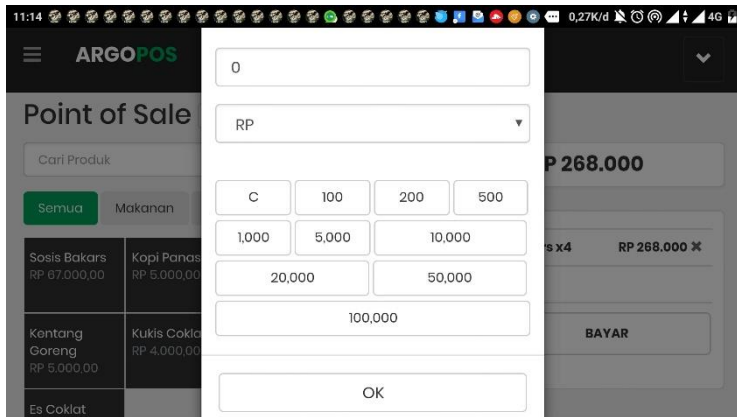
8      success: function(data) {
9          var obj = new Array (
10              "#total", "#total1", "#cartList",
11              "cartList1", "#cart-modal", "#mobile-footer"
12          );
13          reload(obj);
14      },

```

Gambar 5.41 Kode setelah sukses dihapus

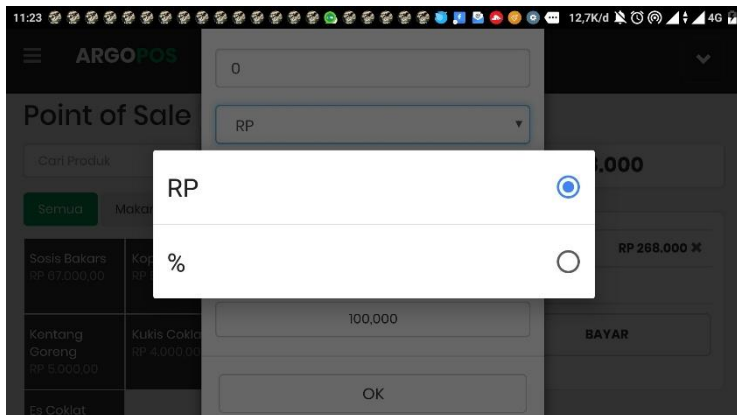
### 5.3.11. Point of Sale: Fungsi Menambahkan Diskon

Fungsi pada *point of sale* selanjutnya adalah untuk menambahkan diskon atau potongan harga pada keranjang belanja. Diskon disini berlaku untuk seluruh barang dalam keranjang belanja dan bukan per produk. Diskon yang dimasukkan dapat dengan satuan Rupiah (RP) atau persen (%). Tombol diskon terdapat pada sebelah tombol bayar dengan *icon tag*, Gambar x. Tombol diskon dapat digunakan setelah setidaknya ada satu barang pada keranjang belanja. Jika tombol diskon diklik maka akan memunculkan *modal* untuk *input* jumlah diskon seperti pada Gambar 5.42



Gambar 5.42 *Pop-up* input diskon

Pada *modal input* terdapat kolom untuk memasukkan jumlah diskon dan satuan diskon berupa *dropdown* apabila diklik akan menampilkan pilihan seperti pada Gambar 5.43.



Gambar 5.43 Pilihan jenis / satuan diskon

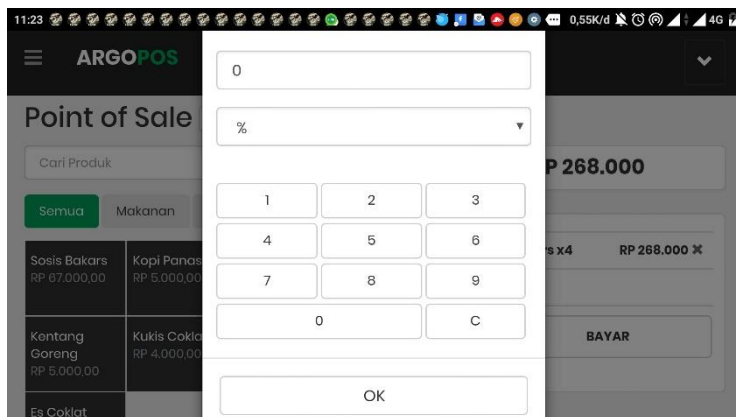
Dan juga terdapat tombol yang memudahkan pengguna untuk memasukkan angka dengan tanpa harus membuka *virtual keyboard* dan harus menunggu muncul di layar. Tombol yang ditampilkan ada dua jenis, yang pertama, pada Gambar 5.42, adalah tombol jika yang dipilih adalah satuan Rupiah (RP), maka tombolnya terdiri dari angka dengan jumlah sesuai nilai mata uang Rupiah yang beredar sehingga tinggal memilih, dan jika setelah mengklik satu tombol dan kemudian klik tombol lainnya, maka nilainya akan dijumlahkan, kode dari tombol tersebut ada pada Gambar 5.44, dimana ketika salah satu tombol diklik akan mengambil nilai yang ada pada kolom jumlah kemudian menghilangkan tanda koma yang menjadi pemisah nilai ribuan dan diubah menjadi *integer*, kemudian mengambil nilai dari tombol yang diklik dan diubah ke *integer*. Kemudian keduanya dijumlahkan dan diubah kembali ke

```

1 $(document).on('click', '#rpbutton .btn-rp .btn-nom', function() {
2   var x = parseInt($("#jumlah-diskon").val().replace(/,/g, ""));
3   var y = parseInt($(this).attr('value'));
4   var z = (x+y).toString().replace(/(\d)(?=(\d\d\d)+(!\d))/g, "$1,");
5   $("#jumlah-diskon").val(z);
6 });

```

Gambar 5.45 Kode untuk tombol angka pada satuan rupiah



Gambar 5.44 Input diskon dengan satuan persen

*string* dan diberikan tanda koma untuk menjadi pemisah nilai ribuan dan dimasukkan nilainya ke kolom jumlah diskon.

Yang kedua, pada Gambar 5.45, jika yang dipilih adalah dengan satuan persen (RP), maka tombol yang muncul adalah tombol seperti pada *Numpad* yang terdiri dari angka 0 hingga 9, yang jika diklik dua tombol atau lebih nilainya bukan dijumlahkan melainkan digabung (*concat*) seperti pada kalkulator, kode dari tombol tersebut ada pada Gambar 5.46. Jika kolom jumlah masih bernilai nol, maka pertama akan dikosongkan, supaya ketika tombol diklik, nilai nol tidak ikut digabungkan di depan. Kemudian mengambil nilai di kolom jumlah dan nilai dari tombol yang diklik kemudian digabung / *concat*. Dan jika nilai yang digabungkan berjumlah lebih dari 100 maka nilai akan tetap sama seperti nilai yang ada pada kolom atau tidak terjadi penggabungan nilai.

Dan untuk menghapus nilai yang sudah dimasukkan, keduanya sudah disediakan tombol *clear* (C).

```

1 $(document).on('click', '#percentbutton .btn-pr .btn-nom',function() {
2   if ($("#jumlah-diskon").val()==0) {
3     $("#jumlah-diskon").val("");
4   }
5   var x = ($("#jumlah-diskon").val());
6   var y = $(this).attr('value');
7   var z = (x+y);
8   if (parseInt(z)>100) {
9     z=x;
10  }
11  $("#jumlah-diskon").val(z);
12 });
13 .

```

Gambar 5.46 Kode untuk tombol angka pada satuan persen

Setelah memasukkan satuan dan jumlah diskon, pengguna klik tombol OK, yang kemudian akan dijalankan dengan kode *jQuery* pada Gambar 5.47, untuk mengirimkan nilai dari diskon ke

```

1 $(document).on('click', '#btn-diskon',function() {
2   var diskon=parseInt($("#jumlah-diskon").val().replace(/,/g, ""));
3   var jenis=$("#jenis-diskon").val();
4   $.ajax({
5     type: "POST",
6     url: "discount.php",
7     data: {diskon:diskon,jenis:jenis},
8     dataType: "JSON",
9     success: function(data) {

```

*discount.php*, menggunakan AJAX, supaya tidak memuat ulang seluruh halaman penuh.

Data yang dikirimkan ke *discount.php* adalah nilai dari jumlah diskon dan juga jenis diskon. Pada *discount.php*, pertama akan dijalankan *query select* pada tabel *orders\_info*, Gambar 5.48, untuk mengecek apakah sudah ada row oleh pengguna.

```
13 $sql = "SELECT * FROM orders_info
14     WHERE user_id = '".$_SESSION["id"]."'";
15
```

Gambar 5.48 Kode *query* untuk mendapatkan data transaksi

Jika *row* dari pengguna sudah ada, maka akan dijalankan *query* untuk memperbarui nilai pada *row* yang sudah ada, Gambar 5.49, yang diperbarui adalah nilai dari jumlah diskon dan jenis diskonnya.

```
18 if (mysqli_num_rows($result) > 0) {
19     $sqlupdate = "UPDATE orders_info
20         SET discount = '".$_POST['diskon']."' , discount_type = '".$_POST['jenis']."'
21         WHERE user_id = '".$_SESSION["id"]."'";
22     mysqli_query($conn, $sqlupdate);
23 }
24
```

Gambar 5.49 Kode *query* untuk memperbarui data

```
25 else {
26     $sqlinsert = "INSERT INTO orders_info (user_id, discount, discount_type)
27         VALUES ('".$_SESSION["id"]."', '".$_POST['diskon']."', '".$_POST['jenis']."'");
28     mysqli_query($conn, $sqlinsert);
29 }
30
```

Gambar 5.50 Kode *query* untuk memasukkan data diskon

Dan jika *row* dari pengguna belum ada, maka akan dijalankan *query* untuk menambahkan data baru, Gambar 5.50, berisi jumlah diskon dan jenis diskon yang dimasukkan.



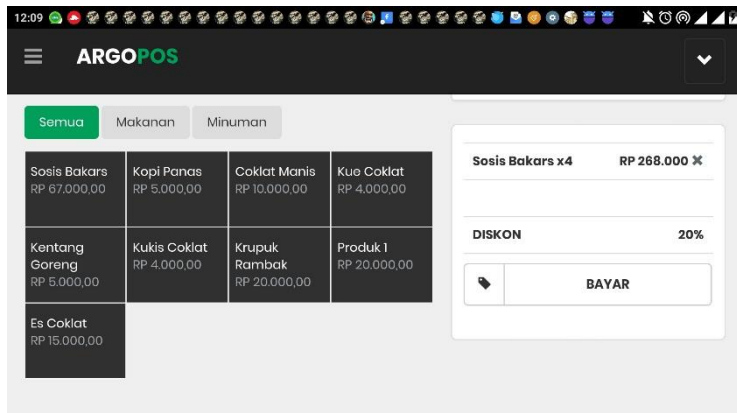
Setelah informasi diskon dimasukkan, maka bagian yang berhubungan seperti panel total, dan keranjang belanja akan dimuat ulang dengan kode pada Gambar 5.51 untuk memperbarui kontennya. Keranjang belanja yang ditambahkan informasi diskon akan tampak seperti pada Gambar 5.52.

```

9      success: function(data) {
10          var obj = new Array ("#row-diskon", "#row-diskon1", "#total", "#total1");
11          reload(obj);
12      },

```

Gambar 5.52 Kode setelah memasukkan informasi diskon



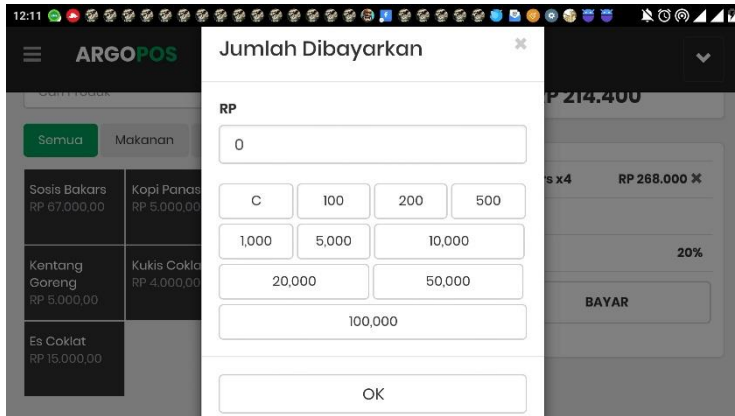
Gambar 5.51 Tampilan informasi diskon pada keranjang belanja

### 5.3.12. Point of Sale: Fungsi Pembayaran

Fungsi ini adalah untuk melakukan pembayaran setelah semua barang yang dibeli oleh pelanggan selesai dimasukkan ke keranjang belanja. Fungsi pembayaran dapat dijalankan setelah setidaknya ada satu barang di dalam keranjang belanja. Tombol bayar ada pada bagian paling bawah pada keranjang belanja. Tombol bayar apabila diklik akan memunculkan *modal*, Gambar 5.53, dengan dengan kolom untuk pengguna memasukkan jumlah tunai yang dibayarkan oleh pelanggan. Seperti ketika memasukkan diskon, disini juga

disediakan tombol sesuai pecahan uang Rupiah yang ada untuk memudahkan pengguna memasukkan jumlah tunai tanpa harus menggunakan *virtual keyboard* pada layar ponsel yang mungkin mengganggu penggunaan aplikasi.

Kode penjumlahan tombol-tombol nominal dan memasukkannya



Gambar 5.53 *Pop-up* input pembayaran

ke kolom jumlah menggunakan kode *jQuery* yang sama dengan fungsi penjumlahan pada fungsi penjumlahan pada tombol diskon pada Gambar 5.53 sebelumnya. Setelah jumlah tunai pembayaran dimasukkan, kemudian diklik OK, dan akan dijalankan dengan kode *Javascript* untuk mengirimkan inputan ke *payment.php* dengan menggunakan *AJAX*, seperti pada Gambar 5.54.

```

1  $(document).on('click', '#btn-bayar', function() {
2      var total = '<?php echo $tot; ?>';
3      var diskon = $('#total').text().replace(/\.|RP /g, '');
4      var paid = $('#jumBayar').val().replace(", ", "");
5      $.ajax({
6          type: "POST",
7          url: "payment.php",
8          data: {total:total,diskon:diskon,paid:paid},
9          dataType: "JSON",

```

Gambar 5.54 Kode untuk tombol bayar

Data yang dikirimkan selain jumlah tunai pembayaran juga jumlah total sebelum diberikan diskon dan total harga setelah diskon. Pada *payment.php*, pertama akan men-*select* pada tabel *orders\_info* untuk mengecek apakah sudah ada *row* oleh pengguna.

```

15  $sql = "SELECT * FROM orders_info
16  : : : : WHERE user_id = '". $_SESSION["id"]. "'";
17

```

Gambar 5.55 Kode *query* untuk mendapatkan data transaksi

Jika *row* dari pengguna sudah ada, maka akan dijalankan *query* untuk memperbarui nilai pada *row* yang sudah ada, Gambar 5.56, yang diperbarui adalah nilai dari total harga sebelum diskon, tunai yang dibayarkan, dan nilai uang kembalian yang didapat dari mengurangi tunai dibayarkan dengan total harga setelah diskon.

```

20  if (mysqli_num_rows($result) > 0) {
21      $sqlupdate = "UPDATE orders_info
22      : : : : SET total = '". $_POST['total']. "',
23      : : : :      paid = '". $_POST['paid']. "',
24      : : : :      changes = '($_POST['paid'] - $_POST['diskon'])
25      : : : :      WHERE user_id = '". $_SESSION["id"]. "'";
26      mysqli_query($conn, $sqlupdate)
27  }

```

Gambar 5.56 Kode *query* memperbarui data transaksi

Dan jika *row* dari pengguna belum ada, maka akan dijalankan *query* untuk menambahkan data baru, Gambar 5.57, berisi total harga sebelum diskon, tunai yang dibayarkan, dan nilai uang kembalian.

```

29  else {
30      $sqlinsert = "INSERT INTO orders_info (user_id, total, paid,
31      : : : : changes)
32      : : : : VALUES ('". $_SESSION["id"]. "', '". $_POST['total']. "'
33      : : : :      , '". $_POST['paid']. "', '". ($_POST['paid'] - $_
34      : : : :      POST['diskon']). "')";
35      mysqli_query($conn, $sqlinsert)
36  }

```

Gambar 5.57 Kode *query* memasukkan data pembayaran

Setelah informasi pembayaran dimasukkan, maka keranjang belanja dan tombol pembayaran akan dimuat ulang dengan kode pada Gambar 5.58 untuk memperbarui kontennya. Keranjang belanja yang ditambahkan informasi pembayaran akan tampak seperti pada Gambar 5.59, ditampilkan jumlah tunai yang

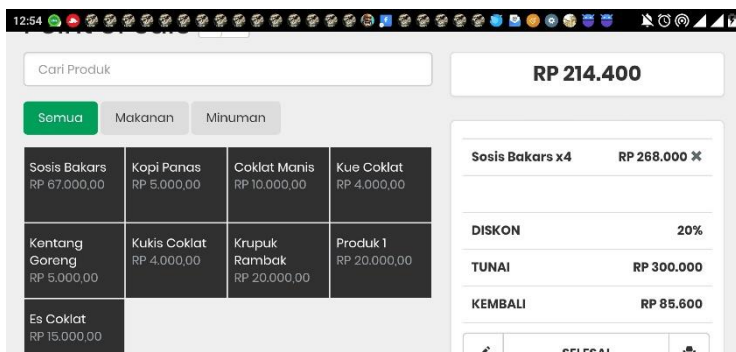
```

10      success: function(data) {
11          var obj = new Array ( "#ket1", "#ket2", "#ket1-1", "#ket2-1",
12                                "#btn-final", "#mobile-footer");
13          reload(obj);
14      },

```

Gambar 5.58 Kode setelah memasukkan data pembayaran dibayarkan dan jumlah kemaian.

Pembayaran tersebut tadi belum akhir dari proses pembayaran, jumlah tunai yang dibayarkan masih dapat diedit jika ada kesalahan. Dan nilai yang dimasukkan ke basis data masih belum dimasukkan ke tabel penjualan. Untuk menyelesaikan pembayaran dapat diklik tombol selesai untuk menghapus data pada tabel *orders\_info* dan *orders\_temp* dan menambahkan data penjualan ke tabel *orders* dan *orders\_detail*, yang diproses juga dengan *Javascript* dan *AJAX*



Gambar 5.59 Tampilan informasi pembayaran pada keranjang belanja

pada Gambar 5.60, yang mengirimkan data total harga sebelum diskon ke *payment-final.php*.

```

1  ✓ $(document).on('click','#done',function() {
2      var total=$("#total-hid").text();
3      ✓ $.ajax({
4          type: "POST",
5          url: "payment-final.php",
6          data: {total:total},
7          dataType: "JSON"
8      });

```

Gambar 5.60 Kode untuk tombol selesai

Pada *payment-final.php*, terlebih dahulu didefinisikan beberapa variabel yang diperlukan, Gambar 5.61, untuk memasukkan informasi ke basis data, termasuk mendapatkan nilai dari tabel *orders\_info* milik pengguna yang melakukan transaksi untuk

```

6  $selecInfo = "SELECT * FROM orders_info
7      WHERE user_id = '". $user_id ."'";
8  $resInfo = mysqli_query($conn, $selecInfo);
9  $rowInfo = mysqli_fetch_assoc($resInfo);
10 $date = date("ymd");
11 $user_id = $_SESSION["id"];
12 $umkm_id = $det['umkm_id'];
13 $total = $_POST['total'];
14

```

Gambar 5.61 Kode definisi variabel yang digunakan dipindahkan ke tabel *orders*.

Selanjutnya menentukan jumlah diskon, apabila satuannya menggunakan Rupiah (RP), maka jumlah diskonnya sama dengan jumlah diskon yang dimasukkan, jika satuannya persen (%), maka jumlah diskonnya adalah total harga dikalikan dengan jumlah diskon yang dimasukkan dibagi 100.

```

15 if ($rowInfo['discount_type']=='RP') {
16     $disc = $rowInfo['discount'];
17 }
18 else {

```

Gambar 5.62 Kode menentukan nilai berdasar satuan

Kemudian informasi yang sudah didefinisikan dimasukkan dengan *query insert* ke tabel *orders*, Gambar 5.63.

```

22 $insertOrder = "INSERT INTO orders (user_id, umkm_id, total,
    discount, paid, changes)
23 VALUES ('".$user_id."', '".$umkm_id."', '".$total."
    ', '".$disc."', '".$rowInfo['paid']."', '".$
    rowInfo['changes']."' );";

```

Gambar 5.64 Kode *query* memasukkan data transaksi

Selanjutnya, data yang baru saja dimasukkan ke tabel *orders*, di-*select* untuk mendapatkan id-nya untuk membuat *order\_code* baru yang digunakan untuk menghubungkan tabel *orders* dan *orders\_detail* karena id yang di-generate otomatis dengan *auto increment* tidak akan cocok antara kedua tabel jika data ini nanti

```

27 $order = "SELECT * FROM orders WHERE user_id = '$user_id' ORDER BY
    id DESC LIMIT 1";
28 $orderResult = mysqli_query($conn, $order);
29 $rowOrder = mysqli_fetch_assoc($orderResult);
30
31 $order_code = $rowOrder['user_id'] . "/" . $rowOrder['umkm_id'] . "/"
    . $date . "/" . $rowOrder['id'];
32

```

Gambar 5.63 Kode mendefinisikan kode penjualan

diimpor ke basis data *remote*.

Kode penjualan dibuat dengan format [id pengguna] / [id UMKM] / [tanggal] / [id]. Setelah membuat kode penjualan, kemudian membuat *query select*, Gambar 5.64, untuk mendapatkan data barang yang ada pada keranjang belanja untuk dipindah dari

```

33 $cart = "SELECT product, sum(qty) as sub FROM orders_temp

```

Gambar 5.65 Kode *query* mendapatkan data di keranjang belanja

*orders\_temp* ke *orders\_detail*, Gambar 5.66, dan memasukkan kode penjualan ke tabel *orders\_detail* dan tabel *orders*.

```

39 while($row = mysqli_fetch_assoc($result)) {
40     $insertDetail = "INSERT INTO orders_detail (order_id, product,
        umkm_id, user_id, qty)
41         VALUES ('".$order_code."', '".$row['product']."', '
            ".$umkm_id."', '".$user_id."', '".$row['sub']."'
            ";
42     $insertCode = "UPDATE orders
43         SET order_id = '".$order_code."',
44         WHERE id = '".$rowOrder['id']."'";
45     mysqli_query($conn, $insertDetail);
46     mysqli_query($conn, $insertCode);
47 }

```

Gambar 5.66 Kode *query* memasukkan informasi detail penjualan

Terakhir adalah *query* untuk menghapus data tabel *orders\_temp* dan *orders-info*, Gambar 5.67, yang berisi informasi dari keranjang belanja milik pengguna, termasuk data barang yang dibeli, informasi diskon dan pembayaran.

```

50 $deleteTemp = "DELETE FROM orders_temp WHERE user_id = '".$user_id."'";
51 $deleteInfo = "DELETE FROM orders_info WHERE user_id = '".$user_id."'";
52

```

Gambar 5.67 Kode *query* menghapus data sementara

Setelah seluruh *query* pada *payment-final.php* selesai dijalankan semua, kembali melanjutkan kode *Javascript* dan menjalankan fungsi *reload()*, Gambar 5.68, untuk memuat ulang keranjang belanja dan panel total harga untuk dikosongkan seperti awal.

```

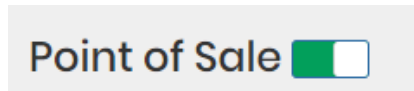
9     var obj = new Array("#total", "#total1", "#cartList",
10     "#mobile-footer", "#cartEmpty");
11     reload(obj);
12 });
13

```

Gambar 5.68 Kode setelah proses selesai dilakukan

### 5.3.13. Point of Sale: Pengaturan Gambar

Pengaturan gambar yang ada pada halaman *point of sale* berfungsi untuk mengganti tampilan produk dari tampilan dengan gambar menjadi tampilan tanpa gambar dan sebaliknya, sehingga ketika aplikasi dijalankan secara online dengan menggunakan koneksi yang lambat dapat diubah tampilannya menjadi tanpa gambar supaya lebih ringan ketika dimuat. Tombol *toggle* pengaturan dapat ditemukan di atas halaman disamping tulisan *Point of Sale*, Gambar 5.69, dan kode untuk fungsi *toggle* pengaturan gambar ditunjukkan



Gambar 5.69 Lokasi tombol *toggle*

pada Gambar 5.70.

Ketika tombol *toggle* diklik akan didefinisikan variabel *setting* apakah *true* atau *false* sebagai *string* kemudian dikirim ke *pic-*

```
1  $("#toggle-event").change(function() {
2      if ($(this).prop('checked') === true) {
3          var setting = 'true';
4      }
5      else {
6          var setting = 'false';
7      }
8      $.ajax({
9          type: "POST",
10         url: "pic-set.php",
11         data: {setting:setting},
12         dataType: "JSON"
13     });
14     reload("#parent");
15     $('#category li').removeClass("active");
16     $('#category li').first().addClass("active");
17 });
```

Gambar 5.70 Kode fungsi tombol *toggle*

*set.php*, Gambar 5.71, dengan AJAX untuk mengganti *session setting* sesuai dengan nilai dari tombol *toggle* tadi, jika *true* berarti



```

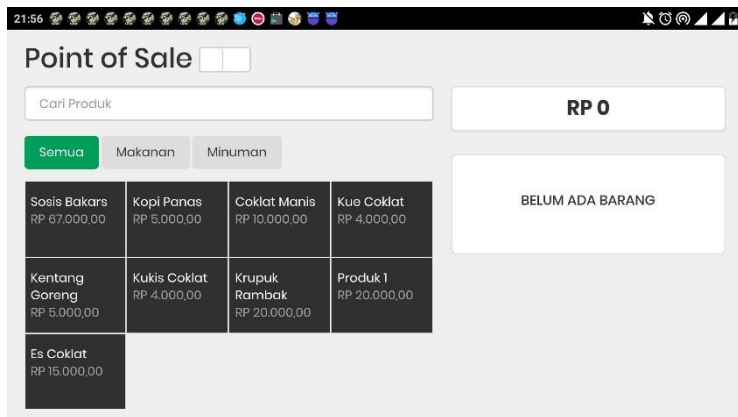
1  <?php
2  session_start();
3  if ($_POST["setting"]!=null) {
4      $_SESSION["setting"] = $_POST["setting"];
5  }
6

```

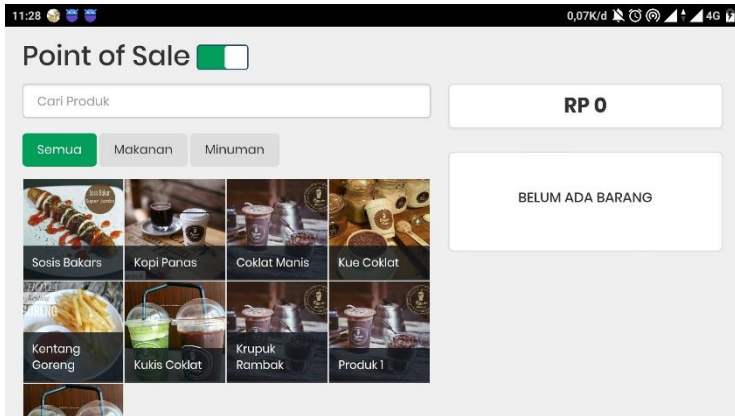
Gambar 5.71 Kode mengeset *setting*

gambar akan ditampilkan, dan jika *false* maka gambar tidak akan ditampilkan. Dan sebelum tombol *toggle* diklik atau diubah, *session setting* setelah login akan secara *default* diatur sebagai *false* dan gambar tidak ditampilkan.

Untuk tampilan halaman *point of sale* dengan gambar ada pada Gambar 5.72, dan tampilan dengan tanpa gambar ada pada Gambar 5.73.



Gambar 5.72 Tampilan dengan tanpa gambar



Gambar 5.73 Tampilan dengan gambar

#### 5.3.14. Point of Sale: Pencarian Produk

Merupakan fungsi untuk dapat mencari produk berdasarkan nama produk dengan mengetikkan nama produk pada kolom pencarian untuk mempermudah pengguna dalam mencari produk yang dibeli oleh pembeli. Kolom pencarian produk dapat ditemukan di bawah judul halaman *point of sale*. Kode untuk pencarian produk ada pada Gambar 5.74.

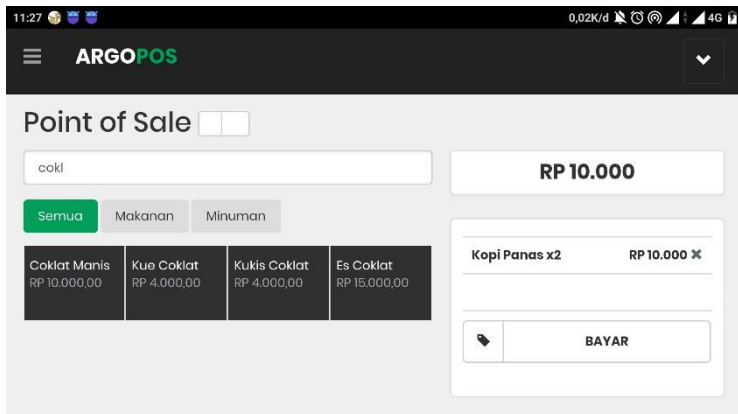
```

1  $(document).ready(function() {
2      $("#search").on('input',function() {
3          var search = new RegExp($(this).val(), 'gi');
4          $(''.box').show().not(function() {
5              return search.test($(this).find(''.name').text());
6          }).hide();
7      });
8  });

```

Gambar 5.74 Kode fungsi pencarian produk

Jadi ketika pengguna mengetikkan huruf pada kolom pencarian akan dijalankan fungsi pada *Javascript*, yang pertama akan didefinisikan variabel sesuai dengan setiap karakter yang diketikkan ke dalam kolom pencarian, dengan atribut *g*, yaitu *global*, untuk menemukan karakter yang sama pada seluruh kata atau kalimat, tidak hanya karakter depan saja, dan atribut *i*, yaitu *insensitive*, untuk mengabaikan besar kecilnya karakter. Selanjutnya akan menyembunyikan konten produk yang di dalamnya terdapat teks yang sama dengan variabel yang didefinisikan, yaitu masukkan setiap karakter yang diketikkan ke dalam kolom pencarian. Pada Gambar 5.75, ditunjukkan contoh ketika pengguna mengetikkan “cokl”, maka pada daftar produk



Gambar 5.75 Tampilan *filter* pencarian sesuai input akan menampilkan produk yang mengandung kata “cokl” dengan mengabaikan besar kecilnya huruf.

### 5.3.15. Point of Sale: Filter Kategori Produk

Merupakan fungsi untuk menyeleksi atau menyaring produk yang ditampilkan pada halaman *point of sale*. Penyaringan dilakukan berdasarkan pada kategori produk, yang dapat ditambahkan pada halaman daftar produk. Tombol untuk penyaringan produk berdasarkan kategori ada di bawah kolom pencarian produk, tombol yang ditampilkan juga sesuai dengan kategori produk yang ditambahkan oleh pengguna. Ketika pengguna mengklik salah satu tombol kategori, maka akan dijalankan kode *Javascript* pada Gambar 5.76.

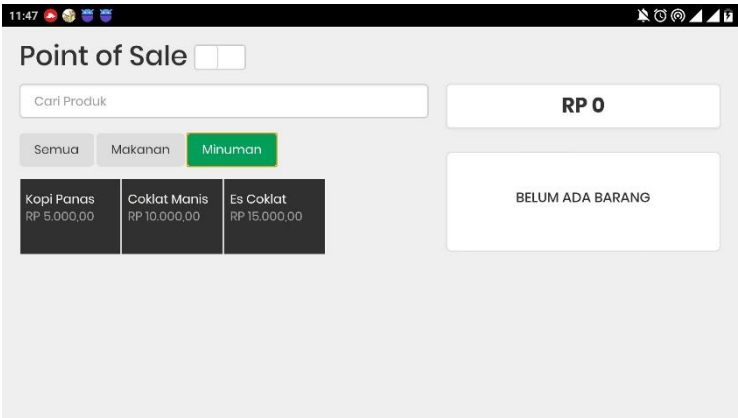
Dengan mengklik salah satu tombol kategori, maka akan didefinisikan variabel untuk membaca kategori yang dipilih dari teks pada tombol yang diklik, kemudian akan disembunyikan

```
1 $(document).ready(function() {  
2     $('#category li').click(function() {  
3         var value = $(this).text();  
4         if(value == "Semua") {  
5             $('#box').show('1000');  
6         }  
7         else {  
8             $(".box").not('.'+value).hide('3000');  
9             $('#box').filter('.'+value).show('3000');  
10        }  
11        $('#category li').removeClass("active");  
12        $(this).addClass("active");  
13    });  
14 }
```

Gambar 5.76 Kode fungsi tombol kategori

produk yang *class* nya tidak sama dengan kategori yang dipilih, dengan menampilkan produk yang memiliki nama *class* sama dengan kategori yang dipilih. Dan jika yang dipilih adalah tombol Semua, maka semua produk akan ditampilkan. Dan setiap mengklik tombol kategori, tombol yang diklik akan ditambahkan *class active* untuk memberikan warna berbeda yang menunjukkan kategori mana yang sedang dipilih, dan juga menghapus *class active* yang sebelumnya aktif pada tombol lainnya. Contoh untuk tampilan penggunaan tombol filter ada pada Gambar 5.77, dimana

pengguna memilih kategori minuman dan ditampilkan beberapa produk saja yang memiliki kategori minuman.



Gambar 5.77 Tampilan setelah tombol kategori diklik

5.3.16. Fungsi Menampilkan Data Penjualan

Fungsi ini adalah untuk menampilkan data penjualan yang telah dilakukan oleh UMKM. Pada setiap *row* data penjualan terdapat tombol Tampilkan untuk menampilkan rincian penjualan, seperti barang yang dibeli, total, diskon, pembayaran, kembalian, dan

Daftar Penjualan

Search..

NO	NO PENJUALAN	TOTAL	DIKSON	TANGGAL	ACTION
1	1/1/180619/188	RP 10,000.00	RP 1,000.00	2018-06-19 20:28:47	Tampilkan
2	1/1/180619/187	RP 0.00	RP 0.00	2018-06-19 20:16:24	Tampilkan
3	1/1/180619/186	RP 0.00	RP 1,000.00	2018-06-19 20:15:21	Tampilkan

Gambar 5.78 Tampilan halaman daftar penjualan

pengguna yang melakukan transaksi. Tampilan dari halaman daftar penjualan dapat dilihat pada Gambar 5.78.

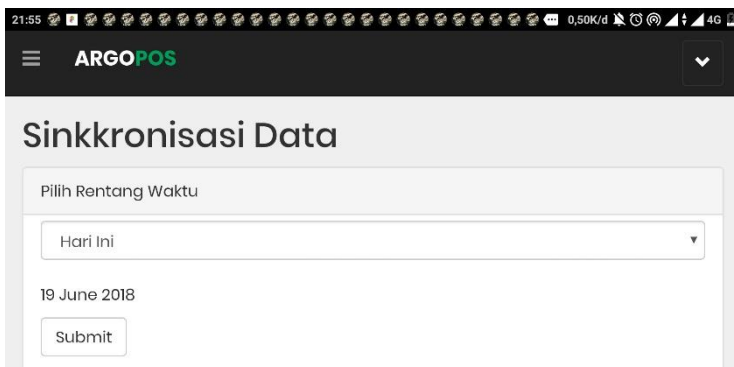


NO	BARANG	HARGA	JUMLAH	SUBTOTAL
1	Rambak Sapi	RP 17,000.00	1	RP 17,000.00
2	Rambak Kerbau	RP 25,000.00	1	RP 25,000.00
3	Kripik Usus Judes Hot	RP 15,000.00	1	RP 15,000.00
HARGA JUAL				RP 57,000.00
DISKON				RP 0.00
TOTAL				RP 57,000.00
TUNAI				RP 70,000.00
KEMBALI				RP 13,000.00

Gambar 5.79 Tampilan halaman detail penjualan

### 5.3.17. Fungsi Sinkronisasi Data Penjualan

Fungsi ini bertujuan untuk mengimpor data penjualan yang dilakukan oleh pengguna yang menjalankan aplikasi pada server lokal ke server *remote* yang menjadi pusat seluruh data penjualan



ARGOPOS

Sinkronisasi Data

Pilih Rentang Waktu

Hari Ini

19 June 2018

Submit

Gambar 5.80 Tampilan halaman sinkronisasi

UMKM. Sinkronisasi data dapat dipilih rentan waktunya, data penjualan satu hari ini saja, minggu ini, atau dapat memilih tanggal awal dan akhir penjualan. Tampilan dari halaman sinkronisasi dapat dilihat pada Gambar 5.79.

Rentang waktu dapat dipilih, hari ini berarti data yang akan diimpor adalah data pada hari ini, minggu ini berarti data pada rentang 7 hari terakhir dari hari ini, pilih tanggal berarti menginputkan sendiri rentang tanggal penjualan, tanggal awal dan tanggal akhir, kemudian untuk mengimpor klik Submit.

Jika diklik *Submit* maka akan diarahkan ke *import.php*, jika diklik tombol pada pilihan hari ini, pada Gambar 5.80, didefinisikan klausa untuk *query select* dengan tanggalnya sama dengan tanggal

```

6  if (isset($_POST["day-submit"])) {
7      $day = $_POST['date'];
8      $clause = "DATE(date) = '." . $day . "'";
9  }

```

Gambar 5.81 Kode jika memilih hari ini

hari ini.

Jika diklik tombol pada pilihan minggu ini, pada Gambar 5.81, didefinisikan klausa untuk *query select* dengan tanggal lebih dari tanggal akhir dan kurang dari tanggal akhir.

```

11 elseif (isset($_POST["week-submit"])) {
12     $start = $_POST['start'];
13     $end = $_POST['end'];
14     $clause = "date >= '." . $start . "' AND date <= '." . $end . " 23:59:59";
15 }

```

Gambar 5.82 Kode jika memilih minggu ini

Jika diklik tombol pada pilihan pilih tanggal, pada Gambar 5.82, didefinisikan klausa untuk *query select* seperti klausa pilihan minggu ini, dengan tanggal lebih dari tanggal akhir dan kurang dari

```

11 elseif (isset($_POST["week-submit"])) {
12     $start = $_POST['start'];
13     $end = $_POST['end'];
14     $clause = "date >= '". $start. "' AND date <= '". $end. "' 23:59:59";
15 }
16

```

Gambar 5.83 Kode jika memilih tanggal

tanggal akhir.

Kemudian pada Gambar 5.84, melakukan *query select* dari tabel *orders* dan tabel *orders\_detail* sesuai dengan klausa yang ditentukan sebelumnya.

```

23 $query="SELECT * FROM orders WHERE ".$clause." AND user_id = '". $id. "'";
24 $query1="SELECT orders_detail.order_id, product, orders_detail.umkm_id,
      orders_detail.user_id, qty, date
25      FROM orders_detail
26      JOIN orders
27      ON orders.order_id = orders_detail.order_id
28      WHERE ".$clause." AND orders.user_id = '". $id. "'";

```

Gambar 5.84 Kode *query* mendapatkan data sesuai klausa

Selanjutnya dari *row* yang di-*select* akan dimasukkan ke dalam dua *array* yang berbeda, seperti pada Gambar 5.85, kemudian pada Gambar 5.86, tutup koneksi pada basis data awal tempat data di-*backup*, dan kemudian dibuat koneksi ke basis data tempat data yang di-*backup* tadi akan disimpan, yaitu pada basis data *remote* di server *online* nya, namun karena yang digunakan untuk uji coba adalah hosting web gratis yang tidak memungkinkan untuk koneksi basis data secara *remote*, maka dari itu dalam uji coba menggunakan basis data yang berbeda pada server lokal.



```

34 $posts = array();
35 while($row = mysqli_fetch_assoc($result)) {
36     $posts[] = array('order_id'=> $row['order_id'],
37                     'user_id'=> $row['user_id'],
38                     'umkm_id'=> $row['umkm_id'],
39                     'total'=> $row['total'],
40                     'discount'=> $row['discount'],
41                     'paid'=> $row['paid'],
42                     'changes'=> $row['changes'],
43                     'date'=> $row['date']);
44 }
45

```

Gambar 5.85 Kode memasukkan data ke *array*

```

57 mysqli_close($conn);
58 $conn = mysqli_connect("localhost", "root", "", "test");
59

```

Gambar 5.86 Kode menutup dan membuka koneksi basis data

Kemudian dari *array* yang telah berisi *row* dari data penjualan tadi akan dimasukkan ke dalam basis data yang berbeda, untuk menyimpan data, lihat Gambar 5.86, dan jika data yang sama sudah ada, maka data akan diabaikan dan tidak dimasukkan.

```

60 foreach ($posts as $key => $val) {
61     $sql = "SELECT * FROM orders
62           WHERE order_id = '". $val['order_id']."'";
63     $result = mysqli_query($conn, $sql);
64
65     if (mysqli_num_rows($result) == 0) {
66         $query = "INSERT INTO orders (order_id, user_id, umkm_id, total,
67                                     discount, paid, changes, date)
68                 VALUES ('". $val['order_id']. "', '". $val['user_id']. "',
69                         '". $val['umkm_id']. "', '". $val['total']. "',
70                         '". $val['discount']. "', '". $val['paid']. "',
71                         '". $val['changes']. "', '". $val['date']. "')";
72         mysqli_query($conn, $query);
73     }
74 }
75

```

Gambar 5.87 Kode memasukkan *array* ke basis data

#### 5.4. Pengujian Aplikasi

Pada bagian ini akan dibahas pengujian yang dilakukan pada aplikasi, mencakup pengujian fungsional sistem dan pengujian kompatibilitas aplikasi terhadap perbedaan ukuran layar.

##### 5.4.1. Pengujian Fungsional

Pada bagian ini dilakukan uji coba terhadap fungsi pada aplikasi yang telah didefinisikan di awal. Setiap fungsi yang ada akan dicoba untuk mengetahui apakah fungsi tersebut sudah berjalan dengan benar dan sesuai dengan yang diharapkan, dan pengujian dilakukan dengan menggunakan sampel data dari studi kasus UMKM Intan Jaya. Pengujian dilakukan dengan menggunakan format *test case* seperti pada Tabel 5.4.

N o	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Kesimpulan
1	Aksi yang dilakukan	Data yang dimasukkan	Hasil yang terjadi pada aplikasi	Hasil yang ditampilkan ke pengguna	Sukses / Gagal

Tabel 5.4 Format *test case*

##### 5.4.2. Pengujian Kompatibilitas

Aplikasi *point of sale* yang dibuat akan diuji kompatibilitasnya terhadap tampilan aplikasi apakah dapat menampilkan halaman dengan benar dan sesuai pada perangkat dengan versi Android yang berbeda. Pengujian dilakukan pada Android versi 5.0, 6.0, dan 7.0, dengan menguji tampilan halaman dengan orientasi *landscape* dan *portrait*.

## **BAB VI**

### **HASIL DAN PEMBAHASAN**

Pada bab ini akan dijelaskan mengenai hasil dari pengujian aplikasi dan pembahasan dari pengujian aplikasi yang telah dilakukan.

#### **6.1. Hasil**

Disini akan dibahas mengenai pengujian yang telah dilakukan terhadap aplikasi.

##### **6.1.1. Pengujian Fungsional Aplikasi**

Hasil dari pengujian fungsional yang dilakukan terhadap aplikasi dapat dilihat pada Tabel 6.1.

Tabel 6.1 Hasil pengujian fungsional

Kode Test Case	Nama Test Case	Status
TC-01	Login	Terpenuhi
TC-02	Register	Terpenuhi
TC-03	Input Produk Baru	Terpenuhi
TC-04	Edit Informasi Produk	Terpenuhi
TC-05	Menghapus Produk	Terpenuhi
TC-06	Input Pegawai Baru	Terpenuhi
TC-07	Edit Informasi Pegawai	Terpenuhi
TC-08	Menghapus Pegawai	Terpenuhi
TC-09	Menambahkan Barang ke Keranjang Belanja	Terpenuhi
TC-10	Menghapus Barang dari Keranjang Belanja	Terpenuhi
TC-11	Menambahkan Diskon	Terpenuhi
TC-12	Pembayaran	Terpenuhi
TC-13	Pengaturan Gambar	Terpenuhi
TC-14	Pencarian Produk	Terpenuhi
TC-15	Filter Kategori Produk	Terpenuhi

TC-16	Menampilkan Data Penjualan	Terpenuhi
TC-17	Sinkronisasi Data Penjualan	Terpenuhi

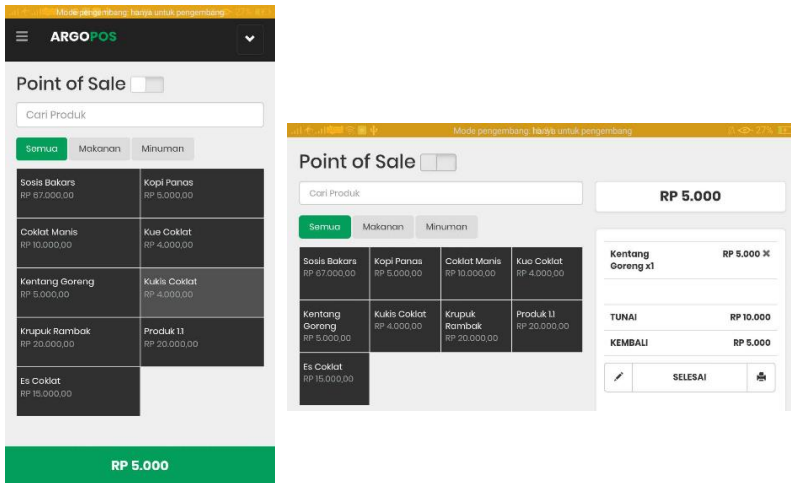
### 6.1.2. Pengujian Kompatibilitas Layar

Aplikasi *point of sale* ini dibangun dengan berbasis *web* yang dirancang untuk dapat responsif ketika digunakan pada perangkat yang berbeda dengan ukuran layar yang berbeda. Sesuai dengan sistem *grid* yang ada pada Bootstrap 3, penyesuaian tampilan halaman aplikasi disesuaikan berdasarkan pada 4 *class*, yaitu **xs** untuk lebar layar dibawah 678px, atau untuk ukuran layar ponsel, **sm** untuk lebar layar lebih dari atau sama dengan 768px, atau untuk ukuran layar tablet, **md** untuk lebar layar lebih dari atau sama dengan 992px, atau untuk ukuran layar *laptop* kecil, dan **lg** untuk lebar layar lebih dari atau sama dengan 1200px, atau untuk ukuran layar *laptop* besar dan *desktop*. Dan selain itu juga ditambahkan *styling* menggunakan CSS untuk menyesuaikan halaman ketika dibuka pada ponsel dengan orientasi *portrait*. Untuk penyesuaian pada keempat *class* sesuai Bootstrap 3 hampir sama semua hanya dengan sedikit penyesuaian ukuran *font* yang digunakan, yang penyesuaiannya berbeda adalah penyesuaian pada layar *portrait*. Untuk perbandingan kompatibilitas ukuran layar pada orientasi layar *portrait* dan *landscape* halaman *point of sale* akan dijelaskan sebagai berikut.

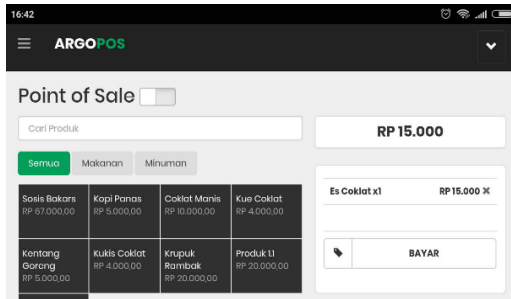
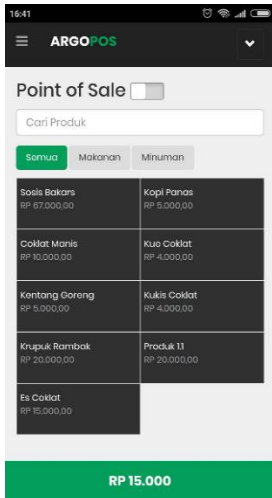
#### a. Daftar Produk

Daftar produk pada halaman *point of sale* memiliki bagian yang paling besar pada orientasi *landscape* karena untuk memudahkan pengguna dalam memilih produk yang ditampilkan, oleh karena itu supaya pada orientasi *portrait* daftar produk dapat ditampilkan dengan ukuran yang cukup besar juga dan halaman *point of sale* juga dapat digunakan dengan mudah, maka untuk orientasi *portrait*, daftar produk diberikan ukuran pada satu halaman penuh meengingat jika

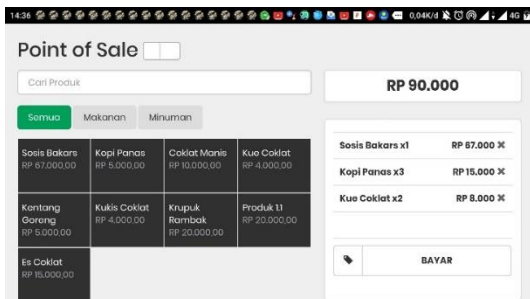
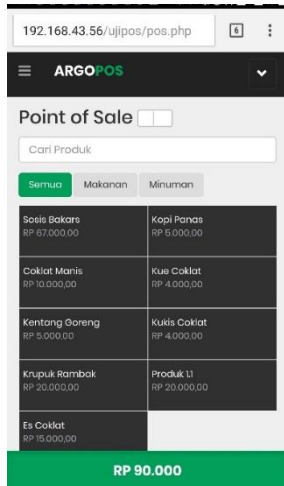
harus diberikan ukuran lebih kecil tampilan daftar produk juga akan lebih kecil dan menyulitkan pengguna ketika memilih produk. Untuk tampilan daftar produk pada Android versi 5.0 dapat dilihat pada Gambar 6.1, untuk Android versi 6.0 pada Gambar 6.2, dan Android versi 7.0 pada Gambar 6.3.



Gambar 6.1 Tampilan daftar barang pada Android versi 5.0



Gambar 6.2 Tampilan daftar barang pada Android versi 6.0



Gambar 6.3 Tampilan daftar barang pada Android versi 7.0

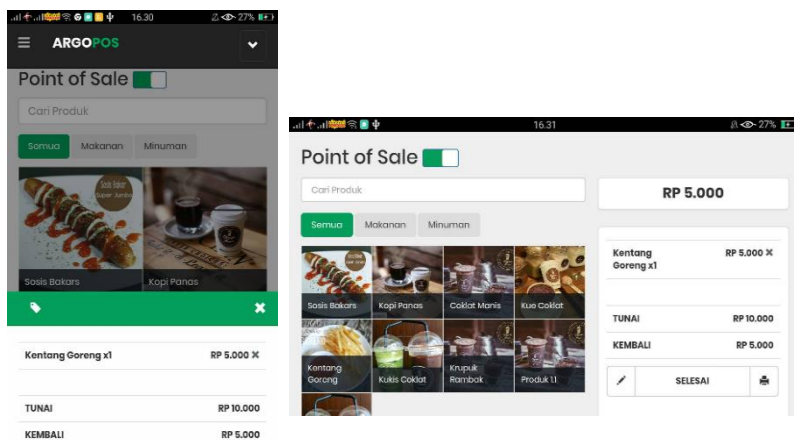
b. Panel Total Harga

Panel total harga pada halaman *point of sale* ditampilkan dengan ukuran yang cukup besar supaya dengan mudah dapat dilihat oleh pengguna. Pada orientasi *landscape* panel total harga berada pada bagian kanan halaman, di sebelah daftar produk dan diatas keranjang belanja, dengan ukuran lebih kecil dibanding daftar produk. Untuk pada halaman orientasi *portrait*, karena halaman sudah penuh dengan daftar produk, maka posisi panel total harga dipindah di bagian bawah halaman dengan posisi *fixed* sehingga ketika halaman digeser keatas atau bawah, panel total harga tetap ada di posisi bawah halaman, untuk tampilannya dan perbandingannya dapat terdapat pada gambar sebelumnya, Gambar 6.1, Gambar 6.2, dan Gambar 6.3.

c. Keranjang Belanja

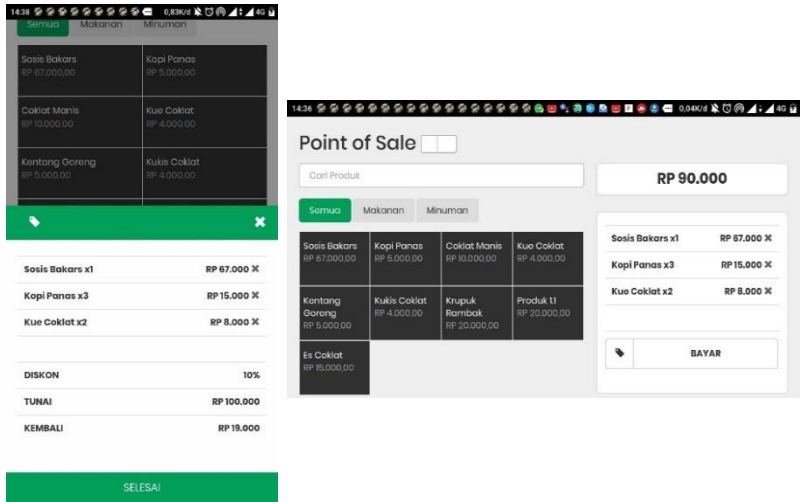
Keranjang belanja pada halaman *point of sale* dengan orientasi *landscape* berada pada bagian kanan halaman dengan porsi yang cukup besar namun dengan ukuran yang lebih kecil dibanding dengan daftar produk. Karena daftar produk akan diakses lebih sering jika dibanding pada keranjang belanja yang hanya diakses pada tombol hapus barang yang lebih jarang dibanding dengan pemilihan produk. Dan ukuran keranjang belanja juga ditampilkan cukup besar supaya pengguna mudah melihat produk mana yang sudah masuk ke keranjang belanja dan informasi pembayarannya. Sedangkan pada orientasi layar *portrait* yang satu halaman sudah dipakai untuk daftar produk, maka untuk keranjang belanja diposisikan berbeda, yaitu berada pada *pop-up* yang akan muncul ketika panel total harga diklik. Karena tidak mungkin untuk meletakkan keranjang belanja disamping daftar produk karena lebar layar dengan orientasi *portrait* yang sempit sehingga tampilan dan ukuran menjadi sulit dilihat, dan akan mempersulit penggunaan, atau diletakkan dibawah daftar produk karena akan mempersulit untuk dapat melihat keranjang belanja karena harus men-*scroll*

halaman ke bawah hingga daftar produk habis, terlebih jika produk yang ditampilkan memiliki jumlah yang banyak. Maka dari itu lebih mudah jika ditampilkan pada *pop-up* yang muncul ketika panel total harga diklik, karena posisi panel yang *fixed* sehingga ketika di-*scroll* ke atas atau ke bawah tombol akan tetap pada posisi di bawah layar dan mudah untuk membuka keranjang belanja, dan tampilan keranjang belanja pada orientasi layar *portrait* pun tidak jauh berbeda dengan tampilan pada orientasi layar *landscape* kecuali pada tombol bayar yang agak berbeda dan tombol diskon yang ada di atas keranjang belanja. Untuk perbandingan tampilan keranjang belanja pada Android versi 5.0 dapat dilihat pada Gambar 6.4, untuk Android versi 6.0 pada Gambar 6.5, dan Android versi 7.0 pada Gambar 6.5.

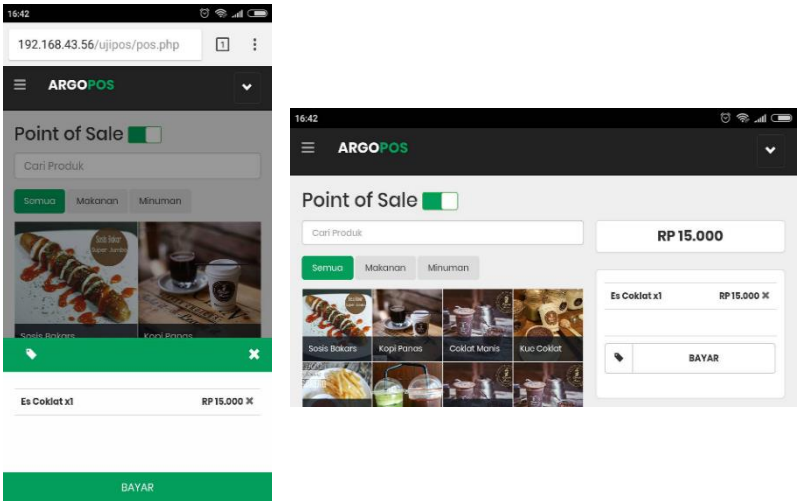


Gambar 6.4 Tampilan keranjang belanja pada Android versi 5.0





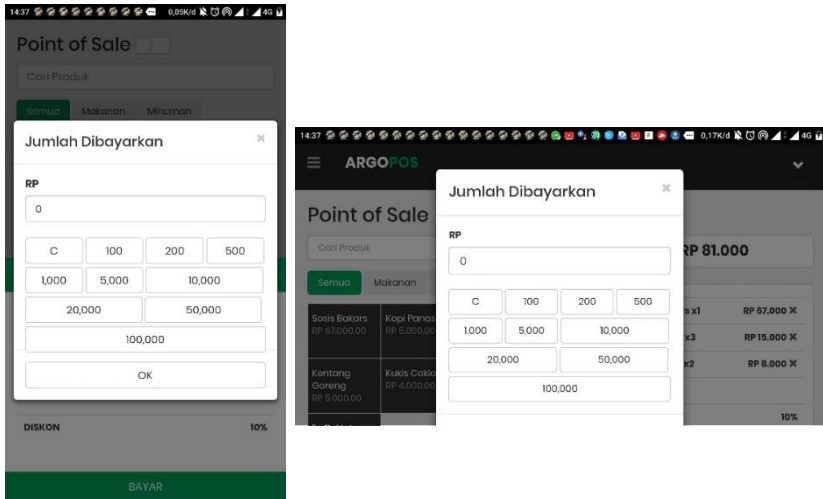
Gambar 6.6 Tampilan keranjang belanja pada Android versi 6.0



Gambar 6.5 Tampilan keranjang belanja pada Android versi 7.0

d. Modal

*Modal* pada halaman *point of sale* pada orientasi *landscape* dan *portrait* memiliki tampilan yang bisa dibilang sama hanya perbedaannya jika pada halaman *landscape*, *modal* muncul di tengah halaman dengan ukuran yang tidak besar, bisa baik untuk *modal* pembayaran dan *modal* diskon, dan pada halaman *portrait*, ukuran lebar *modal* penuh sama dengan lebar halaman, baik untuk *modal* pembayaran dan *modal* diskon. Dan pada orientasi layar yang berbeda pun tombol-tombol angka yang ada tetap tertata sesuai pada tempat seharusnya. Tombol-tombol ini berfungsi untuk mempermudah pengguna dalam melakukan transaksi yang membutuhkan kecepatan, sehingga pengguna tidak perlu menggunakan *virtual keyboard* tiap kali harus menginputkan angka, yang mana biasanya malah mengganggu karena harus menunggu untuk muncul di layar dan apabila performa perangkat menurun juga dapat menyebabkan perangkat berjalan lambat. Dengan adanya tombol angka, pengguna cukup mengklik tombol untuk memasukkan nominal, yang akan dijumlahkan jika mengklik lebih dari satu tombol, dan angka yang tersedia sudah sesuai dengan nominal pecahan rupiah yang sehari-hari digunakan, kecuali pada diskon satuan persen yang berupa bilangan cacah yang apabila diklik lebih dari satu tombol akan menggabungkan angka yang ada dan yang diklik. Pada Gambar 6.7 adalah tampilan dari *modal* pembayaran yang memiliki tampilan tidak jauh berbeda untuk *landscape* atau *portrait* dan untuk versi Android lain pun juga sama.



Gambar 6.7 Tampilan *modal*

Selain dari halaman yang dibahas diatas, lainnya memiliki tampilan layar yang sama pada layar *landscape* maupun *portrait*, karena pada halaman lain tidak terlalu diperlukan dalam penggunaan penuh yang membutuhkan kecepatan dalam menggunakannya seperti pada halaman *point of sale*.

### 6.1.3. Kesimpulan Uji Coba

Kesimpulan dari pengujian yang telah dilakukan adalah sebagai berikut.

1. Pengujian fungsional terhadap fitur yang ada pada aplikasi sudah dapat berjalan dengan baik dan dapat digunakan dengan semestinya sesuai dengan fungsinya masing-masing.
2. Pengujian kompatibilitas terhadap layar pada orientasi *landscape* dan *portrait* pada tiga versi Android yang berbeda, yaitu versi 5.0, 6.0, dan 7.0, menunjukkan jika aplikasi dapat dijalankan pada perangkat dengan versi Android yang berbeda dengan baik, dengan tampilan yang sama dan tata letak yang tertata sesuai pada tempat seharusnya.

3. Dan berdasarkan pada pengujian yang telah dilakukan pada tiga versi Android, 5.0, 6.0, dan 7.0, beserta pengujian pada orientasi layar dengan mode *landscape* dan *portrait* menunjukkan jika aplikasi *point of sale* yang dikembangkan ini sudah sesuai untuk disimpulkan sebagai web responsif, karena sudah dapat menampilkan halaman aplikasi sesuai dengan seharusnya untuk dapat menyesuaikan ukuran pada ukuran dan orientasi layar yang berbeda.

## **BAB VII**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan dijelaskan mengenai kesimpulan yang didapat dari pengerjaan tugas akhir ini dan juga saran untuk pengerjaan tugas akhir ini.

#### **7.1. Kesimpulan**

Dari pengerjaan tugas akhir ini, dapat diambil kesimpulan sebagai berikut.

1. Dengan menerapkan Bootstrap dan CSS pada pengembangan aplikasi, dapat membantu membuat desain tampilan aplikasi untuk dapat menyesuaikan tampilan pada ukuran layar berbeda dan orientasi layar berbeda. Sehingga aplikasi *point of sale* ini dapat digunakan pada perangkat berbeda dan ukuran layar berbeda.
2. Penerapan aplikasi dengan berjalan secara *offline* dan *online* dapat membantu fleksibilitas penggunaan aplikasi yang tidak harus terus menerus terhubung ke jaringan internet, dan data pada basis data lokal dapat disinkronisasikan ke basis data *remote* ketika terhubung dengan jaringan internet.
3. Penggunaan server sendiri di tiap perangkat ponsel dapat menambah fleksibilitas sehingga dapat dijalankan dengan tanpa menggunakan PC dan tanpa perlu terhubung ke jaringan internet.

#### **7.2. Saran**

Berikut adalah saran yang dapat dipertimbangkan untuk tugas akhir ini dapat dikembangkan ke depannya agar lebih baik.

1. Pada aplikasi ini masih dapat dikembangkan untuk dapat melakukan *input* barang ke keranjang belanja dengan menggunakan *scan barcode* menggunakan kamera perangkat atau perangkat *scanner* eksternal.
2. Aplikasi ini dapat dikembangkan lagi supaya dapat diintegrasikan dengan *printer* struk yang biasa digunakan pada

aplikasi POS pada umumnya untuk dapat mencetak bukti transaksi.

## DAFTAR PUSTAKA

- [1] NYC Business Solutions. 2013. How-To Guide:Choosing a POS System. New York.
- [2] Olsera. Olsera POS, 2015 [Online], Available: <https://www.olsera.com> diakses 29 Januari 2018.
- [3] Kounta. Kounta POS, 2012 [Online], Available: <https://www.kounta.com> diakses 29 Januari 2018.
- [4] PT MOKA Teknologi Indonesia. MOKA POS, 2014 [Online], Available: <https://www.mokapos.com> diakses 29 Januari 2018.
- [5] Sugihartono, Jodhi, et al. 2016. Pembuatan Aplikasi *Point of sale* Toko Cabang Perusahaan Torani Menggunakan *Framework* CodeIgniter. Semarang: Universitas Diponegoro.
- [6] Herman, Hendy T., et al. 2015. Pembuatan Aplikasi *Point of sales* untuk Rumah Makan Dapur Rinjani. Surabaya: Universitas Kristen Petra.
- [7] Wijayanto, Hans C. 2016. Penerapan Teknologi *Responsive Web Design* menggunakan Library *Bootstrap* Untuk Pembuatan Aplikasi E-Journal pada Yayasan Bina Darma Salatiga. Salatiga: Universitas Kristen Satya Wacana.
- [8] NYC Business Solutions. 2013. How-To Guide:Choosing a POS System. New York.
- [9] Kyrnin, Jennifer. 2015. Sams Teach Yourself *Responsive Web Design* in 24 Hours. Indiana: Pearson Education.
- [10] Pete LePage. Dasar-dasar *Web Responsif*, 2017 [Online], Available: <https://developers.google.com/web/fundamentals/design-and-ux/responsive/?hl=id> diakses 24 Desember 2017
- [11] Twitter. CSS, 2010 [Online], Available: <https://getBootstrap.com/docs/3.3/css/> diakses 24 Desember 2017
- [12] Panigrahi, K. K. 2017. *Bootstrap Responsive Web Development*.
- [13] Cimo, Fabio. 2015. *Bootstrap Programming Cookbook*.

- [14] Diki Alfarabi Hadi. *Bootstrap* Part 1 : Pengertian Dan Cara Menggunakan *Bootstrap*, 2016 [Online], Available: <https://www.malasngoding.com/pengertian-dan-cara-menggunakan-Bootstrap/> diakses 26 Desember 2017
- [15] Technopedia. Digital Printing. 2016 [Online], Available: <https://www.techopedia.com/definition/14338/digital-printing/> diakses 1 Januari 2018
- [16] KickWe. Android Web Server, 2017 [Online], Available: <https://kickwe.com/tutorial/download-free-android-web-server/> diakses 29 Januari 2018.



## LAMPIRAN

### A. Tabel Pengujian Fungsional

Tabel 9.1 *Test Case Login*

<b>Test Case Code</b>		TC-01			
<b>Test Case Name</b>		Login			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
1	Memasukkan <i>username</i> dan <i>password</i> pada <i>form login</i>	Username = ijaya	Pengguna dapat login	Pengguna berhasil login dan masuk ke aplikasi	Sukses
		Password = 123456			

Tabel 9.2 *Test Case Registrasi*

<b>Test Case Code</b>		TC-02			
<b>Test Case Name</b>		Registrasi			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
2	Pengguna memasukkan data	Nama UMKM = Intan Jaya	Akun pengguna dapat	Akun pengguna didaftarkan ke	Sukses

	pada <i>form register</i> untuk mendaftarkan akun	Nama Pemilik = Masfufah	didaftarkan jika <i>username</i> atau <i>email</i> belum ada	aplikasi dan dapat digunakan untuk <i>login</i>	
		Username = intanjaya			
		Email = masfufah@gmail.com			
		Password = 123456			
		Ulangi Password = 123456			

Tabel 9.3 *Test Case* Input Produk Baru

Test Case Code		TC-03			
Test Case Name		Input Produk Baru			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
3	Pengguna memasukkan data produk yang akan ditambahkan	Nama produk = Rambak Sapi	Data produk disimpan ke basis data	Data produk baru ditambahkan dan ditampilkan di daftar produk	Sukses
		Gambar = file gambar			
		Harga = 17000			
		Kategori = Makanan			
		Keterangan = Rambak sapi produksi intan jaya			

Tabel 9.4 *Test Case* Edit Informasi Produk

<b>Test Case Code</b>		TC-04			
<b>Test Case Name</b>		Edit Informasi Produk			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
4	Pengguna mengubah salah satu informasi produk yang sudah ada	Nama produk = Sanghai Eka -> Shanghai Eka	Informasi produk yang diubah disimpan ke basis data	Informasi produk berubah	Sukses

Tabel 9.5 *Test Case* Menghapus Produk

<b>Test Case Code</b>		TC-05			
<b>Test Case Name</b>		Menghapus Produk			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
5	Pengguna mengklik hapus pada produk yang dipilih	id produk = 1	Produk yang dipilih dihapus dari basis data	Produk dihapus dari daftar produk	Sukses

Tabel 9.6 *Test Case* Input Pegawai Produk

<b>Test Case Code</b>		TC-06			
<b>Test Case Name</b>		Input Pegawai Baru			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
6	Pengguna memasukkan data pegawai yang akan ditambahkan	Nama pegawai = Dewi Ayudia	Data pegawai disimpan ke basis data	Data pegawai baru ditambahkan dan ditampilkan di daftar pegawai	Sukses
		Gambar = kosong			
		Username = dewiaiyu			

Tabel 9.7 *Test Case* Edit Informasi Pegawai

<b>Test Case Code</b>		TC-07			
<b>Test Case Name</b>		Edit Informasi Pegawai			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
7	Pengguna mengubah salah satu informasi	Nama pegawai = Dewi Ayudia -> Dewi Ayu	Informasi pegawai yang	Informasi pegawai berubah	Sukses

	pegawai yang sudah ada		diubah disimpan ke basis data		
--	------------------------	--	-------------------------------	--	--

Tabel 9.8 *Test Case* Menghapus Pegawai

<b>Test Case Code</b>		TC-08			
<b>Test Case Name</b>		Menghapus Pegawai			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
5	Pengguna mengklik hapus pada pegawai yang dipilih	id pegawai = 12	Pegawai yang dipilih dihapus dari basis data	Pegawai dihapus dari daftar produk	Sukses

Tabel 9.9 *Test Case* Menambah Barang ke Keranjang Belanja

<b>Test Case Code</b>		TC-09			
<b>Test Case Name</b>		Menambah Barang ke Keranjang Belanja			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
9	Pengguna memilih produk pada daftar dan memasukkan jumlah barang	id produk yang dipilih	Produk yang dipilih masuk ke keranjang belanja	Produk yang dipilih masuk dan ditampilkan	Sukses
		Jumlah = 1			

				di keranjang belanja	
--	--	--	--	-------------------------	--

Tabel 9.10 *Test Case* Menghapus Barang dari Keranjang Belanja

<b>Test Case Code</b>		TC-10			
<b>Test Case Name</b>		Menghapus Barang dari Keranjang Belanja			
<b>No</b>	<b>Aksi Test</b>	<b>Data Test</b>	<b>Hasil yang Diharapkan</b>	<b>Hasil Aktual</b>	<b>Sukses / Gagal</b>
10	Pengguna mengklik ikon hapus pada salah satu barang di keranjang belanja dan konfirmasi Iya	id barang yang dipilih	Barang dihapus dari keranjang belanja	Barang dihapus dari keranjang belanja dan tidak ditampilkan	Sukses

Tabel 9.11 *Test Case* Menambah Diskon

<b>Test Case Code</b>		TC-11			
<b>Test Case Name</b>		Menambah Diskon			
<b>No</b>	<b>Aksi Test</b>	<b>Data Test</b>	<b>Hasil yang Diharapkan</b>	<b>Hasil Aktual</b>	<b>Sukses / Gagal</b>
11	Pengguna mengklik ikon tag pada	Satuan = %	Diskon dapat ditambahkan ke	Diskon ditambahkan ke	Sukses

	keranjang belanja kemudian memilih satuan diskon dan jumlah diskon	Jumlah = 10	keranjang belanja	keranjang belanja dan total harga berubah sesuai diskon	
--	--	-------------	-------------------	---	--

Tabel 9.12 *Test Case* Pembayaran

Test Case Code		TC-12			
Test Case Name		Pembayaran			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
12	Pemgguna mengklik tombol bayar dan memasukkan jumlah tunai yang dibayarkan kemudian klik selesai	Jumlah = 10000	Data pembayaran dan kembalian dapat ditampilkan di keranjang belanja	Data pembayaran ditampilkan beserta jumlah kembalian, dan keranjang belanja dihapus ketika selesai	Sukses

Tabel 9.13 *Test Case* Pengaturan Gambar

Test Case Code		TC-13			
Test Case Name		Pengaturan Gambar			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
13	Pengguna mengklik tombol <i>toggle</i> untuk mengubah tampilan dari tanpa gambar menjadi dengan gambar dan sebaliknya	Setting = true	Produk yang ditampilkan dapat berubah dari tanpa gambar menjadi dengan gambar	Produk yang ditampilkan berubah dari tanpa gambar menjadi dengan gambar	Sukses

Tabel 9.14 *Test Case* Pencarian Produk

Test Case Code		TC-14			
Test Case Name		Pencarian Produk			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal



14	Pengguna mengetikkan kata pada kolom pencarian	Input = cokl	Produk yang ditampilkan akan menampilkan produk yang memiliki nama mengandung kata “cokl”	Produk yang ditampilkan adalah produk yang memiliki nama mengandung kata “cokl”	Sukses
----	--	--------------	---	---	--------

Tabel 9.15 *Test Case Filter* Kategori Produk

<b>Test Case Code</b>		TC-15			
<b>Test Case Name</b>		Filter Kategori Produk			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
15	Pengguna mengklik tombol kategori yang diinginkan untuk disaring	Kategori = makanan	Produk yang ditampilkan akan menampilkan produk yang memiliki kategori Makanan	Produk yang ditampilkan adalah produk yang memiliki kategori Makanan	Sukses

Tabel 9.16 *Test Case* Menampilkan Data Penjualan

Test Case Code		TC-16			
Test Case Name		Menampilkan Data Penjualan			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
16	Pengguna mengklik tombol <i>detail</i> pada salah satu data penjualan	Id = 6/1/180620/193	Akan ditampilkan detail penjualan dari penjualan yang dipilih	Ditampilkan detail penjualan dari penjualan yang dipilih	Sukses

Tabel 9.17 *Test Case* Sinkronisasi Data Penjualan

Test Case Code		TC-17			
Test Case Name		Sinkronisasi Data Penjualan			
No	Aksi Test	Data Test	Hasil yang Diharapkan	Hasil Aktual	Sukses / Gagal
17	Pengguna memilih rentang waktu data penjualan yang akan	Start date = 2018-06-04	Data penjualan yang dipilih dapat	Data penjualan yang dipilih disinkronisasikan	Sukses

	diimpor, kemudian klik <i>Submit</i>	End date = 2018-06-06	disinkronisasi ke basis data <i>remote</i>	ke basis data <i>remote</i>	
--	--------------------------------------	-----------------------	--	-----------------------------	--

\*) Basis data yang digunakan untuk diekspor data menggunakan basis data yang berbeda pada server lokal, karena server *remote* menggunakan hosting gratis yang tidak memiliki fitur akses *remote* dari server lokal

## ▪ BIODATA PENULIS



Penulis bernama lengkap Bagus Arga Putra lahir di Kabupaten Tulungagung pada tanggal 3 Juni 1996 merupakan anak pertama dari dua bersaudara. Penulis pernah menempuh pendidikan di TK Kartika Wijaya V/28, SD Negeri 5 Kampungdalem, SMP Negeri 2 Tulungagung, dan SMA Negeri 1 Kedungwaru. Di tahun 2014 penulis masuk sebagai mahasiswa di Departemen Sistem Informasi - Fakultas Teknologi Informasi dan Komunikasi – ITS

(Institut Teknologi Sepuluh Nopember) Surabaya, Penulis selain fokus pada bidang akademik juga aktif dalam beberapa organisasi kemahasiswaan di Departemen Sistem Informasi, di Kajian Sistem Informasi sebagai staff Media Kreatif selama dua periode, serta mengikuti kepanitiaan ISE, Pernah juga melakukan kerja praktek di salah satu startup bidang peternakan di Surabaya yaitu PT Ternaknesia Farm Innovation selama 1,5 bulan. di tahun 2017.

Pada pengerjaan tugas akhir di Departemen Sistem Informasi penulis mengambil bidang minat laboratorium Akuisisi Data dan Diseminasi Informasi dengan topik mengenai pengembangan web responsif dengan Bootstrap yang dikhususkan untuk penggunaan pada ponsel pintar, dan aplikasi yang dikembangkan merupakan aplikasi *point of sale* untuk UMKM. Apabila ada keperluan mengenai tugas akhir ini penulis dapat dihubungi melalui *email* [argputra@gmail.com](mailto:argputra@gmail.com)